

# ROBUST OPTIMIZATION WITH APPLICATIONS IN MARITIME INVENTORY ROUTING

A Thesis  
Presented to  
The Academic Faculty

by

Chengliang Zhang

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of Industrial and Systems Engineering

Georgia Institute of Technology  
May 2015

Copyright © 2015 by Chengliang Zhang

# ROBUST OPTIMIZATION WITH APPLICATIONS IN MARITIME INVENTORY ROUTING

Approved by:

Professor George L. Nemhauser,  
Co-advisor  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Professor Joel Sokol, Co-advisor  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Dr. Myun-Seok Cheon  
Corporate Strategic Research  
*ExxonMobil Research and Engineering*

Professor Shabbir Ahmed  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Professor Alan Erera  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Date Approved: February 19, 2015

*In loving memory of my grandparents.*

## ACKNOWLEDGEMENTS

First of all, I would like to express my sincerest gratitude to my advisors, Prof. George Nemhauser and Prof. Joel Sokol. It was their support, guidance and patience that made this thesis possible. I am deeply grateful to them for their invaluable advice, constant encouragement and all the time they spent on cultivating me to be a researcher. I appreciate every bit of help they gave me along the way.

I would like to thank ExxonMobil researchers, Dr. Myun-Seok Cheon, Dr. Ahmet Keha and Dr. Dimitri Papageorgiou for introducing me to an interesting topic and helping me shape my research directions. I am grateful for their time and collaboration. I would also like to thank ExxonMobil Research and Engineering company for financial support. Special thanks to Prof. Shabbir Ahmed and Prof. Alan Erera for serving on my thesis committee.

I am thankful to all the faculty and staff members in ISyE for their helps throughout my graduate studies. I am also indebted to my fellow students at Georgia Tech. Their friendship made my life in Atlanta much more enjoyable.

I would like to express my gratitude to my family for their unconditional love and support. This thesis would not have been completed without their encouragement. Finally, I would like to thank my girlfriend for her love and for sharing the wonderful years with me.

# TABLE OF CONTENTS

<b>DEDICATION</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>SUMMARY</b>	<b>x</b>
<b>I INTRODUCTION</b>	<b>1</b>
1.1 Robustness in maritime transportation	1
1.1.1 Motivation	1
1.1.2 Literature review	5
1.2 Technical preliminaries	7
1.2.1 Mixed integer linear programming	7
1.2.2 Lagrangian relaxation	9
1.2.3 Optimization under uncertainty	10
1.3 Primary contributions	12
<b>II ROBUST MARITIME INVENTORY ROUTING WITH DELIVERY TIME WINDOWS</b>	<b>15</b>
2.1 Introduction	15
2.2 Problem description	16
2.3 A lagrangian heuristic based on relaxing auxiliary soft constraints	22
2.4 A simulator for evaluating the robustness of solutions	25
2.4.1 Random disruptions	25
2.4.2 A recovery model	27
2.5 Computational results	28
2.5.1 Test instances	28
2.5.2 Simulation tests	30

<b>III ROBUST ROUTING AND FLEXIBLE TIME WINDOW ALLO-</b>	
<b>CATION</b>	<b>35</b>
3.1 Introduction	35
3.1.1 Relevant studies	37
3.1.2 Contributions	38
3.2 Problem description	39
3.3 Mathematical formulations	42
3.3.1 Full model	42
3.3.2 Decomposition models for a two-phase heuristic approach	45
3.3.3 An aggregate model based on ship class	47
3.4 Phase I: Robust routing	48
3.4.1 Slack (idle time) reallocation	48
3.4.2 Separating consecutive deliveries	50
3.4.3 Phase I computational results	50
3.5 Phase II: time window placement	53
3.5.1 Disruption scenarios	53
3.5.2 Construction heuristic	53
3.5.3 Phase II computational results	56
3.6 Iterations: updating routes and time-window placement	59
3.7 Summary	61
<b>IV ROBUST SOLUTIONS TO A SINGLE-ITEM UNCAPACITATED</b>	
<b>LOT-SIZING PROBLEM WITH MACHINE BREAKDOWNS</b>	<b>63</b>
4.1 Introduction	63
4.1.1 Relevant studies	64
4.1.2 Contributions	66
4.2 Problem description and mathematical formulation	66
4.2.1 Problem description and budget of uncertainty	66
4.2.2 A mixed integer programming formulation	67
4.3 Solution approach	69

4.3.1	Notation . . . . .	70
4.3.2	Properties of worst-case scenarios . . . . .	71
4.3.3	Robust optimal solutions . . . . .	79
4.4	Numerical experiments . . . . .	80
<b>V</b>	<b>CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS .</b>	<b>85</b>
	<b>REFERENCES . . . . .</b>	<b>88</b>

## LIST OF TABLES

1	Instances description . . . . .	29
2	Simulation results for instances where the robust solution increases the planning cost but performs significantly better when disrupted. . . .	32
3	Simulation results for instances where the robust solution performs significantly better than the original planning solution, and with no increase in planning cost. . . . .	32
4	Simulation results for instances with no increase in planning costs and little difference in performance when disrupted. . . . .	34
5	Phase II: Cost comparisons for various visit sequences . . . . .	57
6	Phase II: Computational performances of the construction heuristic .	60
7	Cost changes of VS1 solutions in Phase III compared to in Phase II .	61
8	Cost changes of VS2 solutions in Phase III compared to in Phase II .	62
9	Saved inventory amounts and stockout levels (case 1, $w_1 + w_2 \leq q$ ) . .	72
10	Saved inventory amounts and stockout levels (case 1, $w_1 + w_2 > q$ ) . .	73
11	Inventory and stockout status (case 2, $w_1 + w_2 \leq q$ ) . . . . .	73
12	Inventory and stockout status (case 2, $w_1 + w_2 > q$ ) . . . . .	74



## LIST OF FIGURES

1	The OECD Industrial Production Index and indices for the world: Gross domestic product, merchandise trade and seaborne shipments, 1975-2013 (1990 = 100). Source: Review of Maritime Transport 2014, UNCTAD. . . . .	2
2	International seaborne trade, selected years (Millions of tons loaded). Source: Review of Maritime Transport 2014, UNCTAD. . . . .	2
3	Ship rerouting under a disruption . . . . .	18
4	Extended time window . . . . .	21
5	Simulated cost difference (robust/original) for Instance 4 . . . . .	33
6	Simulated cost difference (robust/original) for Instance 12 . . . . .	33
7	Simulated cost difference (robust/original) for Instance 17 . . . . .	34
8	Slack reallocation . . . . .	49
9	Discount factor for longer travel arcs . . . . .	49
10	Phase I – Travel cost comparison relative to model without robustness strategy . . . . .	52
11	Phase I – Slack day comparison . . . . .	52
12	Types of disruptions and their impact on the network . . . . .	54
13	Phase II: Time window distribution . . . . .	58
14	Cost changes due to machine breakdowns under different base stock levels . . . . .	82
15	Cost changes due to machine breakdowns under different stockout rates	82
16	Cost changes due to machine breakdowns under different demand rates	83
17	Robust cost comparisons for various base-stock levels and uncertainty budgets . . . . .	84

## SUMMARY

In recent years, the importance of incorporating uncertainty into planning models for logistics and transportation systems has been widely recognized in the Operations Research (OR) and transportation science communities. Maritime transportation, as a major mode of transport in the world, is subject to a wide range of disruptions at the strategic, tactical and operational levels. This thesis is mainly concerned with the development of robustness planning strategies that can mitigate the effects of some major types of disruptions for an important class of optimization problems in the shipping industry.

The problem is motivated by an application in the design and negotiation of an Annual Delivery Plan (ADP) involving a single vendor and multiple customers in the Liquefied Natural Gas (LNG) business. The overall ADP planning activity is to develop contractual agreements of delivery plans that specify delivery dates (or time windows) and the corresponding delivery quantities over a long-term horizon. In the first part of the thesis, we study a maritime inventory routing problem with given time windows for deliveries with uncertain disruptions. We propose a Lagrangian heuristic scheme to obtain robust solutions by incorporating soft constraints, whose satisfaction can aid robustness, into the objective function with Lagrange multipliers. By simulating random disruption events, we show that the actual operational costs in case of disruptions can be significantly reduced when robust plans are implemented. In addition, the simulator enables us to determine the cost of achieving the robustness and to generate recovery solutions under various disruption events with different lead times.

In the second part, we study a more general robust maritime inventory routing

problem with time windows, where the length and placement of the time windows are also decision variables. The vendor must simultaneously decide routes for all the vessels and time windows at all the customers. We formulate the problem as a two-stage stochastic mixed-integer program and propose a two-phase solution approach that considers a sample set of disruptions as well as their recovery solutions. In the first phase, we introduce two planning strategies to generate robust routes, and in the second phase, we propose a multi-scenario construction heuristic to obtain good feasible solutions. We also investigate an iterative procedure between updating the routes and re-optimizing the time windows by coupling the Lagrangian heuristic approach proposed in the first part.

Finally, we study a robust single-item uncapacitated lot-sizing problem with backlogging and random machine breakdowns. The objective is to optimize the costs of production, inventory and backlogging against the worst-case scenario. By identifying the solution characteristics of the worst-case disruptions, we show that the optimal solutions to the robust model can be characterized by a set of stationary production plans.

# CHAPTER I

## INTRODUCTION

Seaborne shipping is a major mode of transportation in the world. Over the past forty years, the world seaborne trade has witnessed a rapid development which is interlinked with the growth in the world merchandise trade and GDP (Figure 1). The gross loading quantity reached 9548 million tons in 2013, and has more than doubled since 1980 (Figure 2). Corresponding to the increase in importance of the maritime industry, new challenges arise for the OR and transportation science communities. In recent years, more and more researchers have recognized the importance of incorporating uncertainty into planning models for logistics and transportation systems. However, compared with other major modes of transport such as air and road-based transportation, studies within the sector of maritime transportation, especially from a perspective of robustness scheduling, have received much less attention. To help bridge the gap, the primary contribution of this thesis is the development of robustness planning strategies that can mitigate the effects of some major types of disruptions for a class of optimization problems in the maritime industry. In this chapter, we introduce the topic of robustness in maritime transportation and provide some core technical preliminaries.

### ***1.1 Robustness in maritime transportation***

#### **1.1.1 Motivation**

There are many uncertain factors that may result in delays and lack of fulfillment of plans in the shipping industry. [23] discusses some problems from the shipping industry where robustness plays an important role and categorizes them into strategic, tactical and operational planning problems. At the strategic level, the uncertainties

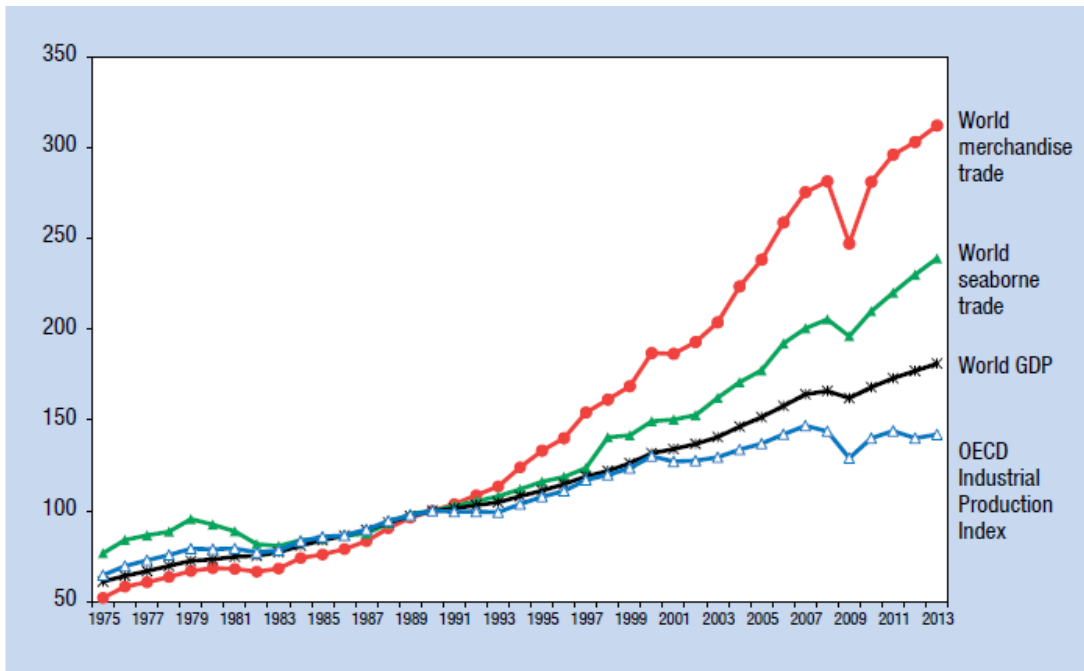


Figure 1: The OECD Industrial Production Index and indices for the world: Gross domestic product, merchandise trade and seaborne shipments, 1975-2013 (1990 = 100). Source: Review of Maritime Transport 2014, UNCTAD.

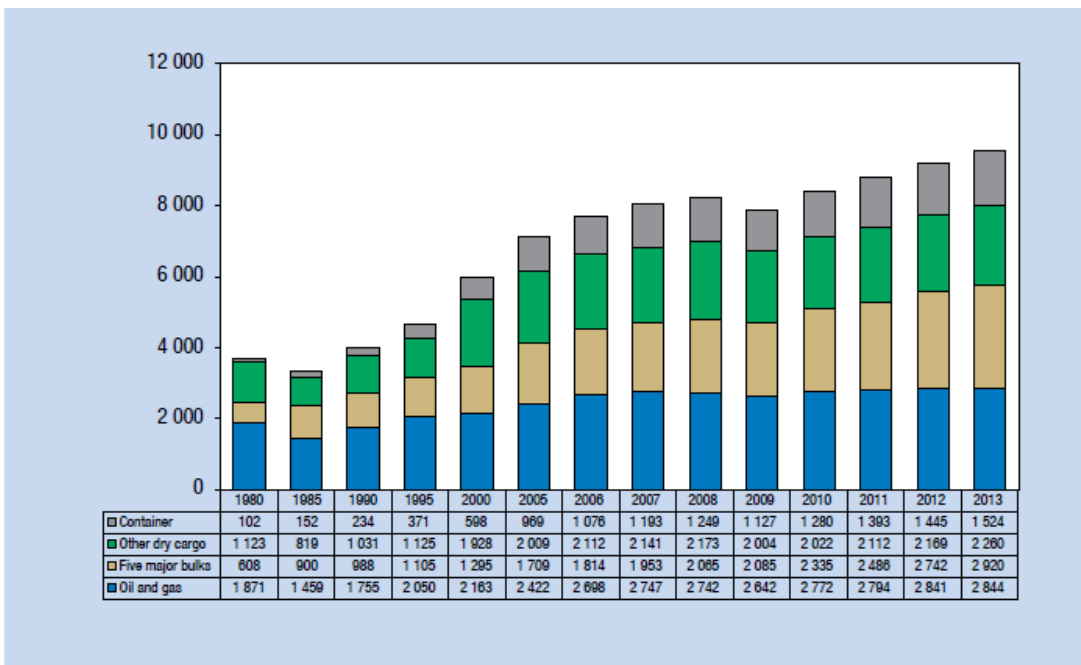


Figure 2: International seaborne trade, selected years (Millions of tons loaded). Source: Review of Maritime Transport 2014, UNCTAD.

can affect the quality of decisions regarding fleet sizing and composition. At the tactical level, they state that “several unpredictable factors influence the fulfillment of plans and should be considered in the planning process. The two most important are probably: (1) weather conditions that can strongly influence the sailing time, and (2) port conditions such as strikes and mechanical problems that can affect the time in port” (p.273). At the operational level, we may consider delays due to tides and restricted open hours at ports.

Decisions at the tactical level consist of ship routing, inventory planning and delivery/berth window scheduling. Specifically, ship routing refers to the assignment of a number of ports in a sequence to a ship; inventory planning determines the amount of loading or discharging of a ship at a port; delivery/berth window scheduling determines the discharging time of a ship at a discharging port. This thesis is mainly concerned about dealing with the disruptions at the tactical level. Depending on the locations of the disruptions, they can be categorized into port delays and en-route delays.

- *Port delays* or *terminal delays* refer to the disruptions that occur during the interval between the arrival and departure of a vessel at a port. Labor strikes are a significant source of port delays and may cause great losses. For example, in December 2014, nearly 108.4 million U.S. dollars was lost due to a one-day strike by the Maritime Workers Union of Nigeria. Multiple parties including the Nigeria Customs Service, licensed customs agents, shipping companies, terminal operators and transporters suffered from the loss. Another major source of port delays is equipment breakdown. According to APM Terminals, there were 3455 crane breakdowns at terminals in 2009 which resulted in more than 600 hours of downtime during operation.
- *En-route delays* or *travel delays* refer to the disruptions that occur between

the departure and arrival of a vessel at two different ports. En-route delays are usually caused due to unforeseen severe weather conditions such as snow, hurricanes and tornados. In 2005, hurricane Katrina caused serious destruction across the U.S. Gulf Coast, which resulted in significant delays to nearly all the ships inbound and outbound to the ports along the coast. Since over 500 million tons of cargo a year, which is nearly 20% of the total cargo tonnage in the United States, both foreign and domestic, moves through the ports in Louisiana and Mississippi, the hurricane caused a tremendous economic loss in the industry.

The disruptions that we consider in this thesis can affect various aspects of our tactical decisions. First of all, they can affect the selection of routes for ships. In a deterministic setting, we are interested in a feasible plan with the lowest travel cost. However, en-route delays can increase travel times between ports and ultimately affect the deliveries at one or more locations. In order to deal with such uncertainties, we may develop routes possessing robust characteristics that allow for flexible re-routing when a disruption occurs. Secondly, we need to manage the inventory levels under uncertainty. Since some storage limit is always specified for a port, delayed arrivals at a loading port may result in an excessive inventory buildup. On the other hand, a loading port that is located favorably for re-routing might experience inventory depletion after a disruption. Thirdly, berth scheduling is also subject to disruptions, especially port delays. Severe port delays can cause port congestion, and largely prolong the waiting time of ships for discharging. Therefore, in this thesis, we consider a class of optimization problems with integrated decisions and unplanned disruptions. We implicitly assume that minor disruptions that can be recovered from by actions such as speeding up ships are accounted for operationally and thus are not considered in our models. Instead, we focus on disruptions that are measured in days, which might cause major delays or have at least a regional impact on the original schedules.

### 1.1.2 Literature review

Within the OR community, there are numerous studies for inventory routing problems (IRPs) which involve an integrated decision-making process combining inventory management and vehicle routing. Maritime inventory routing problems (MIRPs) are the IRPs in maritime transportation. We refer to [9] and [31] for comprehensive reviews of IRPs, and [24], [23] and [58] for comprehensive reviews of MIRPs. Below we review the studies in transportation scheduling under uncertainty in the maritime, air and road-based transportation industries, respectively.

Despite the importance of incorporating robustness into planning models to handle uncertainties, few efforts have been devoted to developing robust optimization models in maritime transportation. [26] studies a multi-ship pickup and delivery problem with soft time windows. They design robust schedules that are less likely to result in ships staying idle at ports during weekends by imposing penalty costs for arrivals at risky times. Also motivated by uncertainties in maritime transportation, [5] and [6] investigate a vehicle routing problem with time windows where travel times are uncertain and belong to a predetermined polytope. A robust optimization framework is used to find routes that are feasible for all values of the travel times in the uncertainty polytope. Similarly, the robust optimization framework is applied in [7] to solve a multi-period fleet sizing and deployment problem with uncertainty in price and demand. A simulation study for a LNG ship routing problem with uncertainty in sailing time and production rate is presented in [41], and several robustness strategies are discussed in the paper. [77] applies three heuristics to a dynamic and stochastic maritime routing problem, and demonstrate that average cost savings of 2.5% can be achieved by including stochastic information in the model.

More work has been done on stochastic airline scheduling problems. [4] studies a robust aircraft routing problem. By adding extra constraints to eliminate optimal solutions already generated at each iteration, multiple solutions are created and then



evaluated based on a robust measure defined as the potential opportunity of swapping planes. [67] presents an optimization model that reschedules legs and reroutes aircraft by minimizing an objective function involving rerouting and cancellation costs, and evaluates the model using a simulation of airline operations. [68] considers a robust fleet assignment problem and defines the robust measure as the hub connectivity of solutions. They propose two fleet assignment models in which one minimizes total costs with limited hub connectivity, and the other isolates hubs with controlled total costs. The results indicate that solutions embedded with many short cycles perform better in operations. [69] studies a robust airline crew scheduling problem under uncertainty. The computational results from three fleets indicate that the crew schedules obtained from their proposed method perform better in a model with disruptions than the crew schedules found via deterministic methods. [50] presents two approaches to minimize passenger disruptions and achieve robust airline schedules. The first approach involves routing aircraft, and the second involves retiming flight departure times. [71] proposes a crew pairing model with move-up crew count. The model aims to maximize the number of move-up crews, i.e. the crews that potentially can be swapped in operations. Delayed column generation and Lagrangian relaxation are used in solving the model. Moreover, it evaluates various crew schedules by generating random disruptions in a crew recovery model. [73] develops fleet assignment solutions that increase planning flexibility and reduce cost by imposing station purity, which limits the number of fleet types allowed to serve each airport in the schedule. [79] proposes a stochastic integer programming model for the airline crew scheduling problem and develops a branching algorithm to identify expensive flight connections and to find alternative robust solutions. [19] investigates slack allocation approaches for robust airline schedule planning. An aircraft re-routing model, a flight schedule re-timing model, and a block time adjustment model, together with their variants are proposed to generate robust schedules.

Compared with maritime and air transportation, much more effort has been devoted in developing robust solutions for road-based vehicle routing problems. In the existing body of studies, the most considered source of uncertainty is stochastic demands. Since uncertain demand is not a focus of this thesis, we restrict our attention to the problems concerned with stochastic travel times in our review. [51] considers vehicle routing problems with stochastic service and travel times, and present a general branch and cut algorithm for a chance constrained model, a three-index simple recourse model and a two-index recourse model. [49] designs vehicle routes between bank branches on a network with stochastic travel times using a heuristic procedure. [46] proposes embedding a branch-and-cut scheme within a Monte Carlo sampling-based procedure, to solve a stochastic vehicle routing problem with random travel and service times. [72] considers a vehicle routing problem to minimize unmet demand with uncertain demands and travel times. A chance constrained formulation of the problem is proposed and solved by a tabu heuristic. More recently, [42] considers a vehicle routing problem with uncertain travel times. They replace the point estimates of travel times in a scenario by range estimates. For each scenario, the robust routes that protect against the worst case within the given ranges are found, and finally the routes with minimum expected cost over all the scenarios are obtained.

## ***1.2 Technical preliminaries***

In this section, we review some major methodologies that are applied in this thesis. We collect some essential concepts of Mixed-Integer Linear Programming (MIP) and Lagrangian Relaxation in Sections 1.2.1 and 1.2.2, respectively. In Section 1.2.3, we discuss some approaches in dealing with optimization problems under uncertainty.

### **1.2.1 Mixed integer linear programming**

A mixed-integer linear program is an optimization problem given by:

$$\min \quad c^T x + d^T y \tag{1}$$

$$\text{s.t.} \quad Ax + By \geq b. \quad (2)$$

$$x \in \mathbb{Z}_+^m, \quad y \in \mathbb{R}_+^n. \quad (3)$$

where  $c \in \mathbb{R}^m, d \in \mathbb{R}^n, A \in \mathbb{R}^{l \times m}, B \in \mathbb{R}^{l \times n}, b \in \mathbb{R}^l$ . It consists of an objective function (1), a set of linear constraints (2), and variable restrictions (3). By replacing the integrality constraints on  $x$  with  $x \in \mathbb{R}_+^m$ , we obtain a linear programming relaxation, whose optimal value provides a lower bound to the mixed-integer linear program. Numerous studies have been conducted in this area from both theoretical and computational perspectives over the past decades. MIP has been widely used in many real applications, and is one of the most important techniques, if not the most, in the applied OR community. [57] provides a thorough treatment of this subject.

A fundamental approach for solving MIP optimization problems is branch and cut, which involves running a branch and bound algorithm and using cutting planes to tighten the linear programming relaxations. Since the class of MIP optimization problems is  $\mathcal{NP}$ -hard, a large variety of techniques are proposed for solving the problems in addition to the branch and cut algorithm. This extensive subject is beyond the scope of the introduction, so we just briefly review some classes of the techniques that will be applied in the thesis. One of them is primal heuristics. The heuristics are used to find good feasible solutions  $(x, y)$  that satisfy (2) and (3) quickly by orientating themselves on some information about the problem which seems helpful to lead to the desired result ([13]). On the dual objective side, cutting-plane methods are another common technique. By adding valid inequalities to cut off solutions with fractional values, they strengthen the linear programming relaxations. Many of the cutting planes are problem-specific, and can be added by users in the branch and cut procedure. A third approach is branching heuristics. Since the branch and bound algorithm consists of two major ingredients, how to split a problem (branching) and which sub-problem to select next, the branching decision is a significant part of the

algorithm. A variety of branching strategies have been proposed for different problems and proven very effective. For an extensive discussion of this topic, we refer to [3].

### 1.2.2 Lagrangian relaxation

Lagrangian relaxation is a relaxation method that can be used to approximate a difficult MIP problem. Suppose constraints (2) are composed of two sets of constraints

$$A^1x + B^1y \geq b^1, \quad (4)$$

$$A^2x + B^2y \geq b^2, \quad (5)$$

where  $A^1 \in \mathbb{R}^{l_1 \times m}$ ,  $B^1 \in \mathbb{R}^{l_1 \times n}$ ,  $b^1 \in \mathbb{R}^{l_1}$ ,  $A^2 \in \mathbb{R}^{l_2 \times m}$ ,  $B^2 \in \mathbb{R}^{l_2 \times n}$ ,  $b^2 \in \mathbb{R}^{l_2}$  and  $l_1 + l_2 = l$ . We include the  $l_1$  constraints that are easier to solve in (4), and  $l_2$  relatively more complicated constraints in (5). By dualizing constraints (5), we obtain a Lagrangian relaxation problem whose optimal value is a lower bound (for minimizations problems) on the optimal value of the original problem. The bound provided by the Lagrangian relaxation problem is at least as good as the bound given by the linear programming relaxation. The problem of maximizing the Lagrangian function of the dual variables (the Lagrangian multipliers) is the Lagrangian dual problem, which is given by:

$$\max_{\lambda \geq 0} \min_{x, y} c^T x + d^T y + \lambda^T (b^2 - A^2 x - B^2 y) \quad (6)$$

$$\text{s.t.} \quad (3), (4).$$

[34] provides a review of the Lagrangian relaxation method for solving integer programming problems. Stimulated by a wide range of applications in Lagrangian relaxation, the subgradient method and its variants are extensively studied as a common method to solve the Lagrangian dual problem. The subgradient method is an iterative procedure where at the  $k$ -th iteration,  $\lambda^{(k+1)}$  is determined by

$$\lambda^{(k+1)} = \lambda^{(k)} + t_k (b^2 - A^2 x^{(k)} - B^2 y^{(k)}) \quad (7)$$

where  $t_k > 0$  is a step size, and  $x^{(k)}$  and  $y^{(k)}$  are the optimal values of the inner minimization problem at the  $k$ -th iteration.

### 1.2.3 Optimization under uncertainty

Various definitions and approaches for solving optimization problems under uncertainty have appeared in the literature. Among them, robust optimization and stochastic programming are the two methods that have been most extensively studied.

Robust optimization is one modeling framework for dealing with uncertain data in optimization. A notable advantage of robust optimization is that it does not rely on the probability distributions of data or uncertain scenarios, which makes this approach attractive when limited information is available to specify a particular distribution. By assuming an uncertainty set, for example, ellipsoidal or polyhedral, we optimize against the worst-case realization of the data. However, the worst-case assumption of robust optimization can lead to solutions that are too conservative. Tractability results depend on the structure of the nominal problem as well as the class of uncertainty set. For detailed explanations of robust optimization, we refer to [15], [10] and [14].

Another major approach for solving optimization problems under uncertainty is stochastic programming. The objective is to find a solution or policy that is feasible for (almost) all the scenarios and minimizes the expected cost under a probability distribution that governs the occurrence of uncertain scenarios. Stochastic programming is a natural framework for problems involving multi-stage decisions. For instance, a solution of a two-stage stochastic program can be regarded as a first-stage policy and a collection of recourse decisions defining what second-stage action should be taken in response to each random scenario. There are two main difficulties when using the stochastic programming approach. One is that the method relies on a particular probability distribution, while estimating such a distribution is often difficult. For

instance, little historical information is available when we try to forecast the demand of a new product or predict the occurrence of natural events. The other difficulty is that stochastic programming has computational limitations as the number of uncertain scenarios grows. We refer to [70] for a comprehensive discussion of stochastic programming.

Instead of using a multi-stage recourse model, the idea of guaranteeing constraint feasibility with a certain probability is related to an approach called chance constrained stochastic programming. Chance constrained programming is a stochastic optimization framework that can accommodate the cases in which the violation of some constraints can almost never be avoided due to the existence of extreme events. By including a set of *chance constraints* or *probabilistic constraints* of the form of  $P(\tilde{A}z \geq \tilde{b}) \geq p$  into a stochastic program, we regard a solution feasible in a stochastic setting if it satisfies constraints  $\tilde{A}z \geq \tilde{b}$  with some high probability  $p$  ( $\tilde{A}$  and  $\tilde{b}$  are functions of random variables). We can categorize chance constrained stochastic programs by identifying whether the probabilistic constraints are disjoint and whether the random variables are independent. There is not a general solution method for chance constrained stochastic programming models. For an extensive discussion of the topic, see [64].

In addition to the most common approaches mentioned above, some other concepts of robustness are proposed in the literature. [12] first extends the static robust optimization model to an adjustable (multi-stage) model in which some of the variables must be determined before the realization of the uncertain parameters (“non-adjustable variables”), while the other variables can be chosen after the realization (“adjustable variables”). [33] investigates a framework called light robustness which couples robust optimization with a simplified two-stage stochastic programming approach. [52] presents a concept called recoverable robustness which aims to find

solutions that can be recovered by limited means in all likely scenarios. [11] proposes a framework called soft robust optimization by assuming that the probability distribution of the data belongs to a given set.

### ***1.3 Primary contributions***

The primary aim of this thesis is to develop robustness planning strategies for a class of optimization problems in maritime transportation which involve integrated decisions including ship routing, inventory planning, berth scheduling and delivery time-window placement. Despite the fact that various methods have been proposed for dealing with optimization problems under uncertainty, it is not clear how these approaches can be adapted into the models concerned with robust scheduling in the shipping industry. In this thesis, we propose several robustness approaches and evaluate their performances in generating robust solutions for MIRPs with random disruptions.

The Lagrangian heuristic approach proposed in Chapter II is a general framework for dealing with a class of optimization problems under uncertainty. The framework can be applied to any optimization problem concerned with robustness for which we can identify a set of constraints whose satisfaction is not necessary but can promote the robustness of solutions. By incorporating such constraints in the objective function with Lagrangian multipliers and updating the multipliers in the dual space, we obtain a pool of candidate robust solutions for simulation evaluation. In Chapter III, we study a more general MIRP that simultaneously considers robust routes and flexible delivery time windows. We introduce two strategies to generate robust routes in which time buffers are spread among deliveries and consecutive deliveries at a port are separated by at least some minimum number of periods. In addition, we propose a multi-scenario construction heuristic to reduce the solution times in determining good and flexible time windows in our proposed two-stage stochastic programming

model.

From a practical perspective, the methodologies proposed in the thesis can be applied to help a vendor design and negotiate long-term delivery contracts with its customers. In vendor managed inventory problems, customers do not reveal their actual inventory levels or consumption rates to the suppliers. Instead, they enter contractual agreements with their vendors. Each agreement specifies a total delivery quantity over the planning horizon, a list of delivery time windows and associated delivery quantity for each time window. Instead of committing a single day for each delivery, placing delivery time windows might be more favorable for the vendor in the presence of disruptions, but is usually associated with a higher cost. Therefore, from the perspective of the single vendor, the problem involves a combination of decisions regarding not only routing and inventory management, but also time window placement. Motivated by this need, in Chapter III, we first introduce a robust maritime inventory routing problem with time window allocation (MIRPTWA) where the length and placement of the time windows are also decision variables, and propose an integrated solution procedure that leads us to robust solutions with lower expected costs under random disruptions. To the best of our knowledge, this generalized maritime inventory routing problem has not been studied in the literature.

In addition, to evaluate the robustness of solutions, we build a simulator that can generate both travel and port disruptions which might affect one or multiple ships and ports for one or several days. The recovery model allows for flexible rerouting so that any ship that is en-route or at a port can be rerouted or rescheduled to a different destination once a disruption is realized. We also consider the effect of lead time in being able to respond to the disruptions.

In Chapter IV, we study a robust lot-sizing problem with random machine disruptions, and use the traditional robustness notion of optimizing against the worst-case scenario. Although the min-max optimization model is intractable, by investigating



the solution structures, we show that the optimal solutions are contained in a set of stationary production plans.

## CHAPTER II

# ROBUST MARITIME INVENTORY ROUTING WITH DELIVERY TIME WINDOWS

### *2.1 Introduction*

The classical Maritime Inventory Routing Problem with Time Windows (MIRPTW) is to find an optimal routing plan that minimizes the total cost of transportation, while satisfying inventory constraints and contractual delivery constraints. However, in practice, unpredictable disruptions may affect the execution of an optimal deterministic plan. Among all the uncertain factors in maritime transportation, one of the most common ones is that travel times are affected by weather conditions. Focusing on this type of uncertainty, we consider MIRPTW with unpredictable disruptions.

Various definitions and approaches for schedule robustness have appeared in the literature. Robust optimization ([10]) is one modeling framework for dealing with uncertain data in optimization. However, the worst-case assumption of robust optimization can lead to solutions that are too conservative. On the other hand, because of the large number of uncertain scenarios that need to be considered, stochastic programming ([70]) has computational limitations for this class of problems. [33] proposes a general heuristic scheme for robustness called Light Robustness where a set of slack variables is used to measure an estimate of the solution robustness and their sum is minimized in the objective function. In this study, we focus on generating robust solutions with limited vulnerability to unpredictable disruptions, and use a different approach for dealing with the uncertainty. After analyzing problem characteristics that may provide robustness, we quantify them as soft constraints that are

incorporated in the objective function with Lagrange multipliers. We use a subgradient algorithm to find candidate solutions to evaluate. Furthermore, to evaluate the robustness of schedules, we build a simulator that generates disruptions and recovery solutions. By simulating various disruption events, we show that the actual operational costs in case of disruptions can be significantly reduced when robust plans are implemented. To the best of our knowledge, only [16] discusses this kind of approach for dealing with robustness in the literature. They propose a Lagrangian heuristic for solving a robust train timetabling problem. The process collects a set of “Pareto optimal” heuristic solutions, and the robustness of a solution is evaluated by calculating a predefined measure.

The main contributions of this chapter are: (i) a general Lagrangian heuristic scheme to deal with robustness where soft constraints are used to promote solution characteristics that lead to robustness, and (ii) a simulator that evaluates the quality of the solutions found by the Lagrangian heuristic algorithm and determines the cost of achieving the robustness.

The remainder part of the chapter is organized as follows. Section 2.2 provides a description of the problem and the mathematical model formulation. Section 2.3 presents soft constraints that are used to enhance robustness and proposes the Lagrangian heuristic scheme to generate robust solutions. Section 2.4 discusses the simulator, random disruptions and the recovery model. We report computational results in Section 2.5.

## ***2.2 Problem description***

The MIRPTW is motivated by the ADP planning problem in the LNG industry. The overall ADP planning activity is to develop contractual agreements of delivery plans that specify delivery dates (or time windows) and the corresponding delivery quantities. Because customers receive product from numerous sources, they typically

negotiate delivery amounts and delivery times with each of their vendors. Therefore, from the vantage point of a single vendor, final agreements are reached through many rounds of negotiations and discussions with multiple customers. At each iteration of the negotiations and after the final agreements are determined, the vendor must generate routing solutions according to the tentative (or final) agreements to check their feasibility, and examine the operational costs and the robustness of the voyages. The scope of this chapter is limited to developing an optimization framework that generates robust routing solutions with given delivery time windows and quantities.

We assume there is a set of loading ports denoted by  $\mathcal{J}^{\mathcal{L}}$ . The  $i$ -th port has a constant production rate  $p^i$  of a single product per period, an initial inventory level  $I_0^i$  and a specified storage capacity  $F^i$ .  $I_t^i$  tracks the inventory level at port  $i$  at the end of period  $t$ . We consider a set of discharging ports denoted by  $\mathcal{J}^{\mathcal{D}}$  where the  $j$ -th port has a set  $\mathcal{K}_j$  of time windows. For the  $k$ -th time window at discharging port  $j$ , we assume that  $u_{jk}$  and  $v_{jk}$  are the first and the last day in the time window respectively, and  $q_{jk}$  is the committed delivery quantity. We assume the time windows at each discharging port do not overlap with each other. The contractual agreements are to deliver quantity  $q_{jk}$  of product in each specified time window  $[u_{jk}, v_{jk}]$  at each discharging port by using a heterogeneous fleet of ships denoted by  $\mathcal{V}$ , each with some load capacity  $W_v$ . We assume vessel speeds are fixed, and service time is already built into the travel time. Let  $\mathcal{T}$  be the set of periods in the planning horizon and  $\mathcal{J} = \mathcal{J}^{\mathcal{L}} \cup \mathcal{J}^{\mathcal{D}}$  be the set of all ports. We restrict our attention to problems where ships discharge their entire load at one port. However, we do not assume that ships necessarily carry full loads. Therefore, we track the quantities on ships for all periods, and use  $O_t^v$  to denote the quantity on ship  $v \in \mathcal{V}$  after period  $t \in \mathcal{T}$ . Also, we assume that more than one ship may serve the same time window.

The model we consider is an abstraction of real maritime transportation planning problems that focuses on the core difficulty of dealing with robustness. Therefore, we

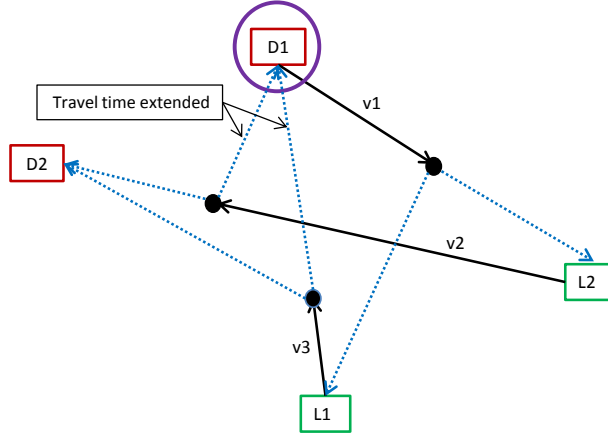


Figure 3: Ship rerouting under a disruption

make some simplifications in the model that do not affect our approach to robustness. For instance, all the ships have the same sailing cost per period and service time at ports. Similar to [66], [76] and [65], we do not include inventory costs in the objective function. One notable difference from most inventory routing and vehicle routing problems considered in the literature is that instead of just providing the distance between each pair of ports, we represent each port by a two-dimensional vector in a coordinate plane. By assuming that ships travel in a straight line between two ports at a constant speed, we can track the geographic locations of ships for all periods. The purpose of doing this is that we are able to adjust the original plan by rerouting some ships en route in case of disruptions. Figure 3 shows an example of rerouting with two loading ports, two discharging ports and three ships. The ships follow the solid lines until a disruption occurs at discharging port D1. Once the disruption is known, each ship can either continue to the original destination or be rerouted to the other loading/discharging port (dash lines). However, the travel times to D1 are extended due to the disruption. Although ships usually do not follow a straight line in reality, we make this assumption to demonstrate our robustness approach while simplifying the physical routing computation. This will be discussed further in Section 2.4.

[74] introduces a practical modeling framework for a class of MIRPs, and the

model we use in this study shares many features with this proposed framework. The model is constructed on a time-space network. The network has a source node  $n_0$ , a sink node  $n_T$  and a set  $\mathcal{N}$  of regular nodes where each regular node  $n$  is a port-time pair  $(j, t)$ ,  $j \in \mathcal{J}, t \in \mathcal{T}$ . The nodes are shared by all the ships, while each ship has its own travel and waiting arcs in the network. The travel arcs from node  $(j_1, t_1)$  to node  $(j_2, t_2)$  represent travel between ports  $j_1$  and  $j_2$ , and the waiting arcs from node  $(j, t)$  to node  $(j, t + 1)$  represent staying at port  $j$  in both period  $t$  and  $t + 1$ . We use  $\mathcal{A}$  to denote the set of all arcs, and  $\mathcal{A}^+$  to denote the set of all travel arcs. In addition, the sets of incoming and outgoing travel arcs associated with ship  $v$  and node  $n = (j, t)$  are denoted by  $\mathcal{RS}(j, t, v)^+$  and  $\mathcal{FS}(j, t, v)^+$  respectively, while the sets of incoming and outgoing arcs associated with ship  $v$  and node  $n = (j, t)$  are denoted by  $\mathcal{RS}(j, t, v)$  and  $\mathcal{FS}(j, t, v)$  respectively.

Let  $x_a = 1$  if arc  $a \in \mathcal{A}$  is used and  $x_a = 0$  otherwise, and  $f_{nv}$  be the loading/discharging quantity at node  $n \in \mathcal{N}$  by ship  $v \in \mathcal{V}$ . An arc-flow mixed integer programming model is given by **(P)**:

$$\min \quad \sum_{a \in \mathcal{A}^+} c_a x_a \quad (8)$$

$$s.t. \quad \sum_{a \in \mathcal{FS}(n, v)} x_a - \sum_{a \in \mathcal{RS}(n, v)} x_a = 0, \quad \forall v \in \mathcal{V}, \forall n \in \mathcal{N} \quad (9)$$

$$\sum_{a \in \mathcal{FS}(n_0, v)} x_a = 1, \quad \forall v \in \mathcal{V} \quad (10)$$

$$\sum_{a \in \mathcal{RS}(n_T, v)} x_a = 1, \quad \forall v \in \mathcal{V} \quad (11)$$

$$I_t^i = I_{t-1}^i + p^i - \sum_{v \in \mathcal{V}} f_{nv}, \quad 0 \leq I_t^i \leq F^i, \quad \forall i \in \mathcal{J}^{\mathcal{L}}, \forall t \in \mathcal{T} \quad (12)$$

$$O_t^v = O_{t-1}^v + \sum_{j \in \mathcal{J}^L} f_{nv} - \sum_{j \in \mathcal{J}^D} f_{nv}, \quad 0 \leq O_t^v \leq W_v, \quad \forall t \in \mathcal{T}, \forall v \in \mathcal{V} \quad (13)$$

$$\sum_{v \in \mathcal{V}} \sum_{u_{jk} \leq t \leq v_{jk}} f_{nv} = q_{jk}, \quad \forall j \in \mathcal{J}^D, \forall k \in \mathcal{K}_j \quad (14)$$

$$f_{nv} \leq W_v \sum_{a \in \mathcal{FS}(j,t,v)} x_a, \quad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \forall v \in \mathcal{V} \quad (15)$$

$$O_t^v \leq W_v (1 - \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a), \quad \forall j \in \mathcal{J}^D, \forall t \in \mathcal{T}, \forall v \in \mathcal{V} \quad (16)$$

$$x_a \in \{0, 1\}, \quad f_{nv} \geq 0, \quad \forall a \in \mathcal{A}, \forall v \in \mathcal{V}, \forall n \in \mathcal{N}. \quad (17)$$

The objective is to minimize total transportation costs, where  $c_a$  is the travel cost associated with arc  $a \in \mathcal{A}^+$ . (9)-(11) are network flow conservation constraints. (12) and (13) are balance constraints of the product at loading ports and ships respectively. (14) ensures that deliveries are completed within the time windows. (15) states that loading or discharging can occur only when the ship is at port. (16) is the full discharge constraint.

Since a ship might have more time than needed to get from one port to another, there can be some slack in planning solutions. Therefore, as a post-processing procedure to improve the robustness of solutions by giving ships as much time as possible for their next voyage, we reallocate the slacks to force ships to depart as soon as possible by using the modified planning model (**MP**):

$$\min \quad g_1(x) = \sum_{a \in \mathcal{A}^+} c_a x_a + \sum_{n=(j,t): j \in \mathcal{J}^D} \varepsilon_n \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{FS}(n,v)^+ \cup \mathcal{RS}(n,v)^+} x_a \quad (18)$$

$$s.t. \quad f_{nv} \leq W_v \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a, \quad \forall j \in \mathcal{J}^D, \forall t \in \mathcal{T}, \forall v \in \mathcal{V} \quad (19)$$

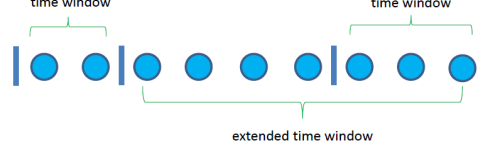


Figure 4: Extended time window

$$(9) - (17)$$

where  $\varepsilon_n$  is an artificial cost whose magnitude is much smaller than the travel cost. Figure 4 shows a discharging port with two consecutive time windows. An extended time is defined as a set of periods that range from the first period after a time window to the last period within the next time window. To model our post-processing stage described above, we set  $\varepsilon_{(j,t_1)} < \varepsilon_{(j,t_2)}$  if  $t_1 < t_2$  and they both belong to the same extended time window.

The second term in (18) together with constraints (19) ensures that ships leave immediately once they collect enough inventory at loading ports and finish discharging at discharging ports. Slack reallocation does not affect the routing decisions since we choose epsilon small enough so that it is dominated by the transportation costs in (18). Moreover, because slack reallocation also breaks symmetry, (MP) is much easier to solve than (P).

Two techniques are applied to speed up solution times. First, we control the order in which variables are branched by specifying a priority order. Specifically in our model, we create a set of ancillary integer variables defined by

$$b_{vjk} = \sum_{u_{jk} \leq t < v_{jk}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a + \sum_{a \in \mathcal{FS}(j,v_{jk},v)} x_a, \quad \forall v \in \mathcal{V}, \forall j \in \mathcal{J}^D, \forall k \in \mathcal{K}_j. \quad (20)$$

By summing the outgoing travel arcs of ship  $v$  over all periods within time window  $k$  at discharging port  $j$ , and the waiting arc of ship  $v$  at the last period within the time window, we define  $b_{vjk}$ , which represents the total number of visits of the ship to the time window. We give the  $b_{vjk}$  variables the highest branching priority.



Secondly, we add a set of enhanced knapsack cover cuts. Since demands are exclusively satisfied from ship deliveries, the inequalities

$$\sum_{v \in \mathcal{V}} W_v b_{vjk} \geq q_{jk}, \quad \forall j \in \mathcal{J}^D, \forall k \in \mathcal{K}_j \quad (21)$$

are valid for the planning model.

### 2.3 *A lagrangian heuristic based on relaxing auxiliary soft constraints*

In this section, we propose a Lagrangian heuristic scheme to obtain robust solutions to our model. The technique is general and can be applied to any optimization problem concerned with robustness for which *soft constraints* can be identified that are not necessary, but whose satisfaction can aid robustness. Since it may not be possible to satisfy all of the soft constraints while satisfying the hard constraints, we incorporate soft constraints in the objective function with Lagrange multipliers. When we increase the Lagrange multipliers by using a subgradient algorithm, various heuristic solutions of potentially higher planning cost are generated.

For problem (MP), suppose  $[u_{jk}, v_{jk}]$  is the  $k$ -th time window at discharging port  $j$ , where  $u_{jk}$  and  $v_{jk}$  are the first and the last period within the time window respectively. An  $m$ -day soft constraint associated with  $[u_{jk}, v_{jk}]$  is defined as

$$\sum_{v \in \mathcal{V}} \sum_{v_{jk}-m+1 \leq t \leq v_{jk}} \sum_{a \in \mathcal{RS}(j,t,v)^+} x_a = 0. \quad (22)$$

It implies that there are no incoming ships in the last  $m$  periods before the end of the time window. Combined with the hard constraint (14), (22) makes all ships that are going to serve time window  $[u_{jk}, v_{jk}]$  get to the port at least  $m$  days in advance. An alternative way of modeling the soft constraints is to replace  $x_a$  in (22) with a continuous variable representing the quantity that ship  $v$  brings to discharging port  $j$  at period  $t$ . Since the two approaches give very similar computational results, for the remainder of the chapter we use the soft constraints defined by (22).

By associating each soft constraint with a multiplier and including  $m$ -day soft constraints ( $m = 1, 2, \dots, r$ ) for all time windows in the objective function, we have the Lagrangian optimization problem:

$$LR(\theta) = \min \{g_2(x) = g_1(x) + \sum_{j \in \mathcal{J}^D} \sum_{k \in \mathcal{K}_j} \sum_{m=1}^r \theta_{jkm} x_{jkm}\} \quad (23)$$

$s.t. \quad (9) - (17), (19)$

where  $x_{jkm} = \sum_{v \in \mathcal{V}} \sum_{v_{jk}-m+1 \leq t \leq v_{jk}} \sum_{a \in \mathcal{RS}(j,t,v)^+} x_a$  and  $\theta_{jkm} \geq 0$  is a Lagrange multiplier.

The dual problem  $\max_{\theta \geq 0} LR(\theta)$  might be unbounded since if all the soft constraints are included the planning problem is likely to be infeasible. Therefore, we impose an upper bound on  $\theta_{jkm}$  for all  $j \in \mathcal{J}^D, k \in \mathcal{K}_j, m \in \{1, \dots, r\}$  and have

$$\max_{0 \leq \theta \leq M} LR(\theta) \quad (24)$$

where  $M = (M^0, \dots, M^0)$  is a vector and  $M^0$  is a large number.

We use a subgradient algorithm to explore the solution space with the step size at the  $k$ -th iteration given by

$$t_k = \frac{\lambda_k(Z^* - LR(\theta^k))}{\|\sum_{j \in \mathcal{J}^D} \sum_{k \in \mathcal{K}_j} \sum_{m=1}^r \theta_{jkm}^k x_{jkm}^k\|^2} \quad (25)$$

where  $\lambda_k$  is a scalar satisfying  $0 \leq \lambda_k \leq 2$ ,  $Z^*$  is an upper bound on problem (24), and  $x^k$  is the optimal solution found at the  $k$ -th iteration. Justification of the formula is given in [44]. The next proposition gives the optimal value of problem (24), and thus can be used as  $Z^*$  in formula (25).

### Proposition 1

$$\max_{0 \leq \theta \leq M} LR(\theta) = LR(M). \quad (26)$$

*Proof:* Let  $\theta^*$  be a Lagrange multiplier such that  $0 \leq \theta_i^* \leq M^0$  for each  $i$ . We assume that  $x$  is an optimal solution of  $LR(\theta^*)$  and  $y$  is an optimal solution of  $LR(M)$ .

Since  $y_{jkm} \geq 0$ ,  $\forall j \in \mathcal{J}^{\mathcal{D}}$ ,  $\forall k \in \mathcal{K}_j$ ,  $\forall m = \{1, 2, \dots, r\}$ , we have

$$\begin{aligned}
& LR(\theta^*) \\
&= g_1(x) + \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{k \in \mathcal{K}_j} \sum_{m=1}^r \theta_{jkm}^* x_{jkm} \\
&\leq g_1(y) + \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{k \in \mathcal{K}_j} \sum_{m=1}^r \theta_{jkm}^* y_{jkm} \\
&\leq g_1(y) + M \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{k \in \mathcal{K}_j} \sum_{m=1}^r y_{jkm} \\
&= LR(M).
\end{aligned}$$

Therefore,  $\max_{0 \leq \theta \leq M} LR(\theta) = LR(M)$ . □

---

**Algorithm 1:** Lagrangian heuristic algorithm for finding robust solutions

---

**0.** Solve  $LR(M)$ . Let  $U = LR(M)$ , and  $x_{(M)}$  be an optimal solution.

**1.** Set  $\theta^0 = 0, k = 0, A = \{x_{(M)}\}$ .

**2.** Solve  $LR(\theta^k)$ . If  $LR(\theta^k) \geq (1 - \epsilon)U$  and  $x^k \notin A$ ,  $A \leftarrow A \cup \{x^k\}$ .

**3.** If  $k = \Omega$  and  $A \neq \{x_{(M)}\}$ , go to 4.

Else if  $k = \Omega$  and  $A = \{x_{(M)}\}$ ,  $A \leftarrow A \cup \{x^\Omega\}$ , go to 4.

Else, update  $\theta^{k+1} = \theta^k + t_k(Ax^k - b)$  where  $t_k = \frac{2(U - LR(\theta^k))}{\|Ax^k - b\|^2}$ ,  $k \leftarrow k + 1$  and go to 2.

**4.** Simulate solution  $x$ ,  $\forall x \in A$ .

---

Now we give the Lagrangian heuristic algorithm for finding robust solutions. In our algorithm, we assume the initial Lagrange multiplier vector  $\theta^0 = 0$  and use set  $A$  to collect candidate robust solutions. We use  $U$  as the upper bound in formula (25), and set  $\lambda = 2$  for every iteration. If the gap between the optimal value of  $LR(\theta^k)$  and  $U$  is no larger than a pre-specified parameter  $\epsilon$ , solution  $x^k$  is labeled as a candidate robust solution.  $\Omega$  is a pre-specified maximal limit on the number of iterations of the subgradient algorithm, and  $k$  denotes the current iteration number. If  $x_{(M)}$  is the only candidate robust solution after the last iteration, we include  $x^\Omega$  (the solution in the last iteration) in set  $A$ . The final output of the algorithm is a set  $A$  that contains

all the different candidate robust solutions that will be processed in the simulator presented in the next section to evaluate their actual performance when disrupted.

## ***2.4 A simulator for evaluating the robustness of solutions***

To evaluate how solutions respond to unexpected disruptions, we built a simulator to study the recovery process from disruptions. The simulator generates random disruptions one by one and reoptimizes the original schedule in each case. To respond to a disruption, we solve a recovery model that incorporates the following three recovery options:

1. *Push-back.*

If the slacks in the schedule or the time windows are sufficient to absorb the delays, we simply delay the affected routes and do not re-route any ships. Push-back does not increase cost.

2. *Ship re-routing.*

If necessary, the simulator is able to re-route ships en route to ports different from their original destinations. This recovery option often increases transportation costs.

3. *Spot market.*

If it is impossible to meet all time window demands by only using the first two options, an expensive spot market acts as an additional supply source in the recovery model.

### **2.4.1 Random disruptions**

A disruption at a loading port is defined as a four-dimensional vector  $(t, p, h, l)$  which means that all the ships that are scheduled to arrive at loading port  $p$  at period(s)  $t, \dots, t + h - 1$  are delayed to period  $t + h$ , and the recovery model is able to respond

to the disruption at period  $t - l$ . A disruption at a discharging port is defined as a triple  $(w, h, l)$  where  $w$  is the affected time window,  $h$  is the number of extra extended travel days to time window  $w$  on ships that are scheduled to serve  $w$ , and  $l$  is the lead time of being able to respond to the disruption before it occurs.

By the definition of a disruption,  $h$  represents the extent of the disruption and  $l$  controls the lead time of when it is possible to respond to the disruption. We now show the obvious result that for a single disruption, the total actual cost is a nonincreasing function of lead time.

**Proposition 2** *Suppose  $c(s, d)$  is the total actual cost over the entire horizon if  $s$  is the planning solution and disruption  $d$  occurs during execution. Assume  $d_1 = (w, h, l_1)$  (or  $d_1 = (t, p, h, l_1)$ ),  $d_2 = (w, h, l_2)$  (or  $d_2 = (t, p, h, l_2)$ ) and  $l_1 > l_2$ . Then  $c(s, d_1) \leq c(s, d_2)$ .*

*Proof:* Assume disruption  $d_1$  becomes known at period  $t_1$  and disruption  $d_2$  becomes known at period  $t_2$  for  $t_1 < t_2$ . The proof would be trivial if at the earlier time  $t_1$ , one could wait until the later time  $t_2$  to take action. However, if rerouting is chosen in our recovery model, it needs to be done immediately at the time of recovery. Consider any ship rerouted in the recovery at time  $t_2$ . Rerouting the ship to the same new destination at time  $t_1$  instead will yield a lower (or equal) cost given the triangle inequality, and will remain feasible because its arrival time at the new destination will be no later. Therefore,  $c(s, d_1) \leq c(s, d_2)$ .  $\square$

Proposition 2 can be used to obtain a lower bound on the actual cost for disruptions with the same location and intensity.

**Corollary 1** *Assume  $c(d^*)$  is the optimal cost of the planning model with the disruption  $d^* = (w, h)$  already known at the start of the planning horizon. Then  $c(d^*) \leq$*

$c(s, d)$ , where  $s$  is any planning solution,  $d = (w, h, l)$  (or  $d = (t, p, h, l)$ ) and  $l$  is any lead time up to the start of the horizon.

*Proof:* It follows immediately from Proposition 2.  $\square$

#### 2.4.2 A recovery model

The recovery model is used to reoptimize over the remaining periods once a disruption occurs. By following a two-stage format of planning and recovery, we focus on responding to a single disruption in the time horizon. So only one disruption is generated in each simulation run. However, real situations may involve sequential disruptions while the original schedule is being executed. Therefore, it is reasonable to include robustness in the recovery model despite the fact that only a single disruption is considered for each scenario. To achieve this, the Lagrangian terms are kept and the strategy of slack reallocation is also applied in the recovery model. The recovery model also includes the proposed recourse options. We use the same notation in the recovery model as in the planning model, but the notation actually represents different sets since we solve a problem with a shorter time horizon in the recovery stage.

$$\min g_2(x) + M' \sum_{j \in \mathcal{J}^D} \sum_{k \in \mathcal{K}_j} s_{jk} \quad (27)$$

$$s.t. \quad \sum_{a \in \mathcal{FS}'(n_0, v)} x_a = 1, \quad \forall v \in \mathcal{V} \quad (28)$$

$$\sum_{v \in \mathcal{V}} \sum_{u_{jk} \leq t \leq v_{jk}} f_{nv} + s_{jk} = q_{jk}, \quad \forall j \in \mathcal{J}^D, \forall k \in \mathcal{K}_j \quad (29)$$

$$(9), (11) - (13), (15) - (17), (19).$$

The objective function (27) includes spot market costs where  $M'$  is a huge number ( $M' > M^0$ ) so that the spot market is only used to avoid infeasibility. (28) is the flow

conservation constraint on the source node, but with  $\mathcal{FS}'(n_0, v)$  defined differently. Specifically, if the disruption becomes known at time  $t - l$ , then

$$\mathcal{FS}'(n_0, v) = \begin{cases} \{n_0 \rightarrow (j_0, 0)\} & \text{if ship } v \text{ is at port } j_0 \text{ at time } t - l, \\ \{n_0 \rightarrow (j, d_j) \mid \forall j \in \mathcal{J}^{\mathcal{L}}\} & \text{if ship } v \text{ is on the route to some loading port at time } t - l, \\ \{n_0 \rightarrow (j, d_j) \mid \forall j \in \mathcal{J}^{\mathcal{D}}\} & \text{if ship } v \text{ is on the route to some discharging port at time } t - l, \end{cases}$$

where  $d_j$  is the distance between the geographic location of ship  $v$  at time  $t - l$  and the port  $j$ . (29) assures that demands are satisfied either from the ship deliveries or the spot market.

## 2.5 Computational results

The dual problem (24) incorporates two robustness strategies: the slack reallocation and the soft constraint approach. By running the simulator, we show their effects on solutions in terms of robustness. Section 2.5.1 gives a description of the instances we use and Section 2.5.2 shows the simulation results.

### 2.5.1 Test instances

A total of 18 instances are created based on the MIRPTW described in Section 2.2. The instances can be categorized into five classes according to the number of loading and discharging ports, and for each instance, a 60-period problem is defined. Depending upon different instances and ports, one-way voyage durations range from 3 to 9 periods. Table 1 provides some detailed information.

Once the simulator is called to evaluate a solution, three types of disruptions are generated in the simulation program:

1. Disruptions at loading ports. A disruption at a loading port is represented as a four dimensional vector  $(t, p, h, l)$ . In the simulation program, we randomly

Table 1: Instances description

Class	C1	C2	C3	C4	C5
Instance No.	1-6	7-12	13,14	15,16	17,18
# of loading ports	2	1	2	2	1
# of discharging ports	2	3	3	3	4
# of ships	3	4	3	4	4
# of time windows at all discharging ports	10-11	12-15	12-13	15	17-18

generate five different pairs of  $(t, p)$ , and then run a total of 30 disruptions that have the form  $\{(t_i, p_i, h, l) : i = 1, \dots, 5, h = 1, 2, 3, l = 2, 5\}$ .

2. Disruptions of a single time window of a discharging port. A disruption at a discharging port is represented as a triple  $(w, h, l)$ . In the simulation program, we enumerate the disruptions over all the time windows  $\mathcal{K}$ , and run a total of  $6|\mathcal{K}|$  disruptions that have the form  $\{(w_{jk}, h, l) : j \in \mathcal{J}^D, k \in \mathcal{K}_j, h = 1, 2, 3, l = 2, 5\}$ .
3. Disruptions of two time windows that are close in time. The purpose is to represent a big disruption that first hits one port and then another. Let  $C = \{(w_{jk}, w_{j'k'}) : |u_{jk} - u_{j'k'}| \leq 5, j, j' \in \mathcal{J}^D, k, k' \in \mathcal{K}_j, (j, k) \neq (j', k')\}$  be a set that defines all pairs of close time windows. In the simulation program, we randomly generate  $6 \min\{5, |C|\}$  disruptions, each involving two close time windows with a 1, 2 or 3-day disruption and a 2 or 5-day lead time.

We set the limit on the maximum number of iterations  $\Omega = 10$  and the gap tolerance  $\epsilon$  between the optimal value of  $LR(\theta^k)$  and  $U$  as 1%. 1-day, 2-day and 3-day soft constraints (22) for all the time windows are included in the Lagrangian optimization model. The integer programs are solved using CPLEX 12.5. The solver is stopped after 3600 seconds (planning model) or 600 seconds (recovery model). If no integer solution has been found, another 3600 or 600 seconds is given, repeating until an integer solution is found.

We simulate these scenarios and solve the corresponding recovery problems in



sequence. The scenarios are sorted such that for disruptions that only differ in extent  $h$  and lead time  $l$ , the ones with larger  $h$  (the first sorting priority) and smaller  $l$  (the second sorting priority) are simulated first. This allows us to use MIPStart features in CPLEX, since any feasible solution for a disruption with larger  $h$  or smaller  $l$  is also feasible for the same disruption with smaller  $h$  or larger  $l$ . Since the timing of disruptions means that we usually solve a problem with a shorter time horizon in the recovery stage, most disruption scenarios are solved to optimality in the first 600 seconds and only a few scenarios in which the disruptions occur at the very beginning of the planning horizon require another 600 seconds.

### 2.5.2 Simulation tests

Tables 2-4 show the results of instances 1 – 18. For every test instance, we select the solution from set  $A$  that has the lowest average simulated cost over all the scenarios and consider it as the robust solution. The cost of the robust solution operated without disruption is called the *robust planning cost* and the optimal cost of the planning model (P) is called the *original planning cost*. In each table, the second column gives the percent cost increase of the robust solution above the optimal planning solution, i.e.

$$\frac{\text{robust planning cost} - \text{original planning cost}}{\text{original planning cost}} * 100.$$

Let

$$AO = \frac{\text{average simulated cost of the original solution} - \text{original planning cost}}{\text{original planning cost}} * 100,$$

and

$$AR = \frac{\text{average simulated cost of the robust solution} - \text{robust planning cost}}{\text{robust planning cost}} * 100.$$

Then the first number in columns 3 – 8 is  $AR$ , and the number in parenthesis is  $AO - AR$ .

The instances are grouped in three tables. For those in Table 2, the robust solution is not the optimal as planning solution. The Lagrangian approach produces a different solution with additional planned travel cost. However, the average simulated cost is reduced significantly when the robust plan is executed. In Figure 5, we provide scenario-by-scenario simulation results for Instance 4 as an example. The scenarios are sorted by the simulated cost improvement of the robust solution over the original solution. There were no instances where the planning solution cost increased without a significant average performance improvement under disruption.

The contrast is even sharper in Table 3, where the robust plans are alternative optimal planning solutions. For some instances, the routing decision of the robust solution is different from that of the original plan, while for the others, their routing decisions are the same but with slacks allocated differently. In Figure 6, we show scenario-by-scenario simulation results for Instance 12. Without paying any price in the planning solution, we improve the robustness of the system significantly.

However, our approach does not make much improvement for the instances shown in Table 4, and sometimes the robust solution is slightly inferior to the original solution. One possible reason is that given all the pre-specified hard time windows, it is likely that there exists no plan that could make all the deliveries robust simultaneously. In this case, a planner has to make tradeoffs among these plans, or in other words, among various deliveries. We use Instance 17 to illustrate this. The scenario-by-scenario simulation results are given in Figure 7. The two solutions perform equally well under more than 85% of the scenarios, but neither dominates for all of the rest. Given the hard constraints, there is always some limitation on the robustness of all the feasible solutions. When the original solution nearly reaches the robustness limitation, our approach usually provides some alternative solutions that perform as well in an average sense, but might emphasize different deliveries in terms

of robustness. In addition, since the recovery model for reoptimizing the robust solutions includes the Lagrangian terms while the recovery model for reoptimizing the original solutions does not, it is likely that under some disruptions, the recovery costs associated with the robust solutions could be slightly higher than the original ones.

Finally, the lead time effect stated in Proposition 2 can be observed in the average simulated cost with 5 days of lead time as opposed to 2.

Table 2: Simulation results for instances where the robust solution increases the planning cost but performs significantly better when disrupted.

Ins.	Planning cost increase	Average simulated cost over all scenarios $AR(AO - AR)$					
		lead time = 2			lead time = 5		
		1 day	1-2 days	1-3 days	1 day	1-2 days	1-3 days
1	10.2	0.0 (15.8)	0.0 (21.5)	4.4 (25.3)	0.2 (10.9)	0.2 (11.9)	4.6 (12.3)
3	1.7	11.1 (24.1)	22.5 (22.9)	29.9 (18.9)	3.9 (11.0)	7.6 (14.3)	13.4 (13.7)
4	3.4	0.0 (22.6)	5.0 (25.6)	6.7 (32.8)	0.0 (5.6)	3.5 (13.5)	5.3 (21.1)
6	1.6	6.9 (15.6)	6.9 (18.8)	15.5 (14.8)	6.9 (11.6)	6.9 (13.1)	12.0 (12.8)
13	4.8	4.4 (11.2)	7.0 (18.1)	10.9 (28.8)	4.4 (10.5)	7.1 (11.9)	10.2 (19.6)
14	5.4	13.5 (35.4)	14.2 (35.8)	20.0 (34.8)	10.5 (6.5)	11.4 (10.2)	17.3 (7.6)
15	1.2	4.4 (17.4)	8.8 (23.2)	15.1 (23.2)	1.4 (13.8)	5.3 (16.8)	11.1 (16.9)

Table 3: Simulation results for instances where the robust solution performs significantly better than the original planning solution, and with no increase in planning cost.

Ins.	Planning cost increase	Average simulated cost over all scenarios $AR(AO - AR)$					
		lead time = 2			lead time = 5		
		1 day	1-2 days	1-3 days	1 day	1-2 days	1-3 days
2	0.0	12.7 (24.7)	27.6 (16.6)	36.9 (12.7)	11.4 (11.6)	21.6 (9.0)	31.3 (6.7)
5	0.0	9.9 (17.5)	12.9 (30.3)	17.3 (31.3)	6.3 (0.5)	9.5 (6.5)	14.3 (12.3)
8	0.0	7.4 (14.5)	9.6 (15.7)	14.6 (15.6)	5.4 (8.8)	7.0 (13.3)	11.8 (12.7)
9	0.0	5.5 (5.7)	7.4 (8.4)	10.1 (14.3)	4.2 (3.1)	6.0 (3.1)	7.0 (7.2)
11	0.0	8.4 (6.0)	11.4 (5.7)	15.1 (9.1)	8.3 (0.4)	10.1 (2.0)	11.9 (7.1)
12	0.0	0.0 (14.0)	3.0 (14.5)	9.2 (17.9)	0.0 (7.9)	1.9 (8.8)	8.4 (12.5)

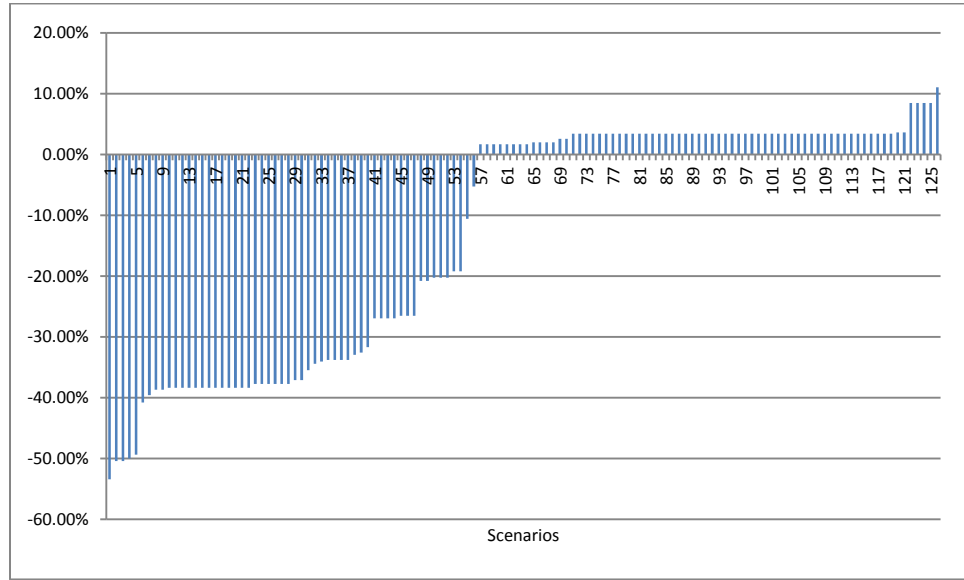


Figure 5: Simulated cost difference (robust/original) for Instance 4

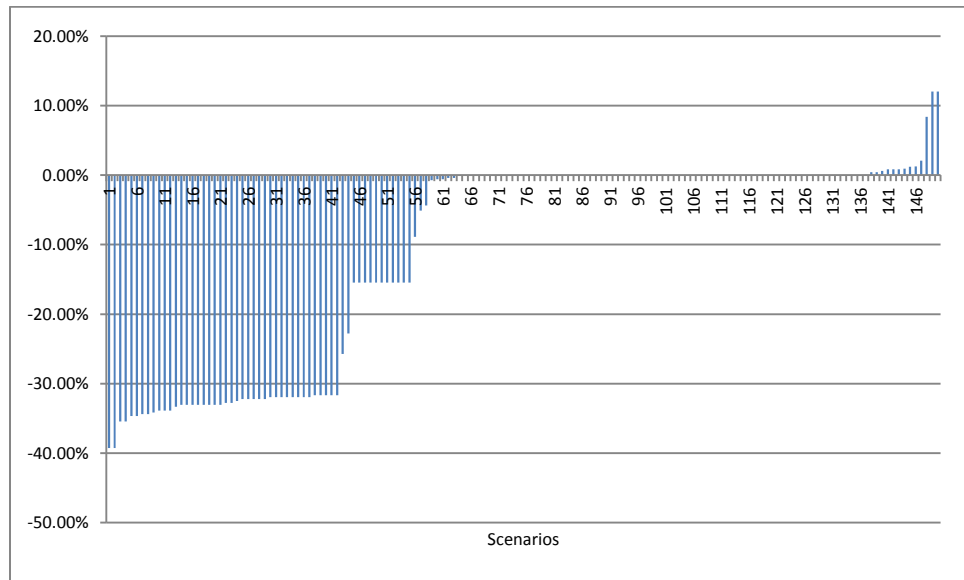


Figure 6: Simulated cost difference (robust/original) for Instance 12

Table 4: Simulation results for instances with no increase in planning costs and little difference in performance when disrupted.

Ins.	Planning cost increase	Average simulated cost over all scenarios $AR(AO - AR)$					
		lead time = 2			lead time = 5		
		1 day	1-2 days	1-3 days	1 day	1-2 days	1-3 days
7	0.0	5.7 (0.0)	7.9 (-0.1)	10.4 (0.0)	1.0 (0.0)	5.3 (0.0)	8.5 (-0.1)
10	0.0	0.0 (0.0)	1.6 (0.0)	7.0 (0.3)	0.0 (0.0)	1.5 (0.0)	6.1 (-1.2)
16	0.0	4.1 (-0.4)	4.1 (-0.4)	10.4 (-0.5)	0.9 (-0.4)	1.3 (-0.5)	6.3 (-0.5)
17	0.0	1.7 (0.0)	4.4 (-0.1)	9.2 (-0.2)	0.5 (0.0)	3.0 (0.0)	7.8 (0.4)
18	0.0	4.9 (0.0)	13.0 (-0.2)	18.6 (-0.1)	1.5 (0.0)	8.8 (-0.1)	14.4 (0.2)

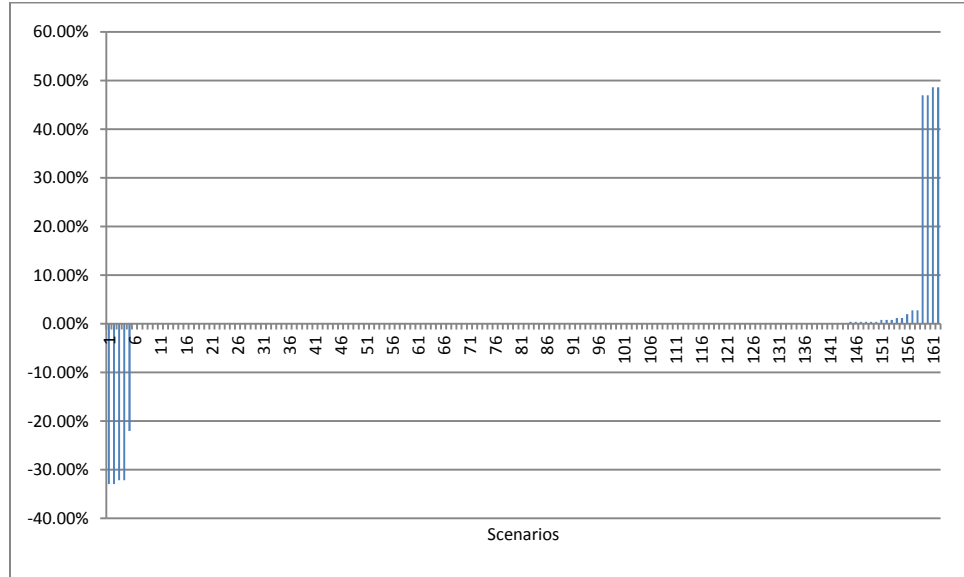


Figure 7: Simulated cost difference (robust/original) for Instance 17

## CHAPTER III

# ROBUST ROUTING AND FLEXIBLE TIME WINDOW ALLOCATION

### *3.1 Introduction*

In this chapter, we study a robust maritime inventory routing problem with time windows for deliveries with uncertain disruptions, where the length and placement of the time windows are also decision variables. In a traditional inventory routing problem, a vendor is responsible for both the inventory management at suppliers and customers, and for the routing of vehicles to pick up and deliver products. A variant is to assume that time windows are present (as given data) to account for the fact that pick-ups and deliveries may only be permitted within pre-specified time intervals (Chapter II). In this chapter, we consider the problem faced by a single vendor who must simultaneously decide routes for all the ships and delivery time windows.

The motivation for this work stems from vendor managed inventory problems where a vendor is responsible for delivering product to several customers over a planning horizon. Because the customers receive product from numerous sources, they typically negotiate delivery amounts and delivery times with each of their vendors. Customers do not reveal their actual inventory levels or consumption rates to the suppliers. Instead, they enter contractual agreements with vendors for deliveries over the planning horizon. Meanwhile, from the vantage point of a single vendor, the problem becomes an inventory routing problem with time windows (IRPTW) in which the time windows are decision variables. The vendor must provide to each individual customer a list of time windows for the entire planning horizon. There is an important tradeoff to be made when generating these time windows. On the one hand,

customers prefer to have small time windows to better plan their day-to-day operations, reduce their inventory levels, and lower their overall risk exposure. On the other hand, the vendor prefers to have large time windows so that it is possible to meet all contractual requirements even in the presence of disruptions in the planned delivery routes.

There are two fundamental ways to better withstand disruptions. First, the vendor can strategically develop routes possessing characteristics that allow for flexible recovery when a disruption occurs. Second, he can judiciously place delivery time windows at customers so that there are more opportunities to satisfy a delivery. However, in the absence of disruptions, both options may incur an additional cost above the optimal deterministic delivery solution. Since the vendor is interested in developing a delivery plan that is both economical as well as robust against uncertain disruptions, it is more favorable to have an integrated solution procedure that simultaneously considers the routing and time window allocation. Therefore, the objective of the robust inventory routing problem with time window allocation considered in this chapter is to determine robust routes for all the vehicles and flexible time windows at all the customers under unknown disruptions, so as to minimize the total expected costs.

An important motivating application of this work is in the creation and negotiation of an ADP in the LNG business. We refer to [39] and [8] for a review of the LNG supply chain. [40] proposes a branch-and-price method and implement different accelerating strategies to solve a LNG inventory routing problem. [66], [76] and [65] study a deterministic ADP problem in the LNG supply chain. The goal is to find an optimal plan for a heterogeneous fleet of ships that delivers two types of products from a single depot to a set of discharging ports such that constraints related to inventory storage at loading ports and contractual obligations at discharging ports are satisfied. Delivery time windows are not considered in their studies.

### 3.1.1 Relevant studies

IRPs involve a combination of inventory management and vehicle routing. [9] and [31] give comprehensive reviews of IRPs. “Several applications of the IRP have been documented. Most arise in maritime logistics, namely in ship routing and inventory management.... Problems arising in the chemical components industry and in the oil and gas industries are also a frequent source of applications in a maritime environment” ([31], p.2). Surveys on maritime inventory routing problems are given by [27] and [60].

The IRPTW is a variant of the IRP in which time windows for pick-ups and deliveries are given exogenously (as data). Early work in the maritime sector includes [28] and [25] who study the Inventory Pickup and Delivery Problem with Time Windows, which involves scheduling of a fleet of ships while respecting the service time windows and inventory restrictions. [32] studies a multi-ship pickup and delivery problem with soft time windows. The objective is to find shipping schedules with significant reduction in transportation cost by introducing soft time windows which can be violated at the expense of paying inconvenience cost. For a similar problem, [26] designs robust schedules that are less likely to result in ships staying idle at ports during weekends by imposing penalty costs for arrivals at risky times. [5] and [6] investigate a vehicle routing problem with time windows (VRPTW) where travel times are uncertain and belong to a predetermined polytope. A robust optimization framework is used to find routes that are feasible for all values of the travel times in the uncertainty polytope. [80] develops a Lagrangian heuristic scheme for generating robust solutions to a MIRPTW with uncertain travel disruptions.

However, the problem of assigning time windows has largely been overlooked in the VRP and IRP literature. To the best of our knowledge, this is the first study to consider an inventory routing problem with time window allocation. [75] considers a related problem, referred to as the time window assignment vehicle routing problem,



which focuses exclusively on demand uncertainty. It consists of determining a time window assignment before demand is known, and finding a vehicle routing schedule for each scenario satisfying the time windows such that the expected costs are minimized. Another related work is the vehicle routing problem with self-imposed time windows (VRP-SITW) studied in [45]. Unlike in the traditional VRPTW where time windows are given exogenously by the customers, in the VRP-SITW, the vendor assigns customers to vehicles, sequences the customers allocated to each vehicle, and sets the time windows in which it plans to serve the customers. The term “self-imposed” refers to the fact that the vendor selects the time windows by itself, independently of the customer, and therefore time windows are treated as endogenous to the routing problem. The work also aims to cope with travel time disruptions by inserting slack time into the schedule. Like the VRP-SITW, our problem treats time window allocation as a decision variable to cope with various disruptions. However, compared with the VRP-SITW, we consider a more complex problem which involves a heterogeneous fleet of ships, a multi-period planning horizon and multiple loading ports. Also, unlike in the VRP-SITW where the length of time windows is given, we treat it as a decision variable in our problem.

### **3.1.2 Contributions**

The main contributions of this chapter are:

1. We introduce a robust MIRPTWA where the length and placement of the time windows are also decision variables. The problem involves a combination of not only inventory management and vehicle routing, but also time window placement. To the best of our knowledge, this generalized maritime inventory routing problem has not been studied in the literature.
2. We formulate the problem as a two-stage stochastic mixed-integer program and

propose a two-phase solution approach that considers various types of disruptions that may affect the travel time or availability of vessels to deliver in certain time periods.

3. We develop a modeling technique that aims to generate robust routes in which time buffers are spread among deliveries and consecutive deliveries at a port are separated by at least a minimum number of periods.
4. We propose a stochastic multi-scenario construction heuristic to reduce the solution times in determining good and flexible time windows.

The remainder of the chapter is organized as follows. Section 3.2 provides a description of the problem and discusses the types of disruptions considered in this chapter. Section 3.3 gives the mathematical model formulation and proposes a two-phase solution approach. Section 3.4 presents two robustness strategies for generating robust routes, and gives the Phase I computational results. Section 3.5 proposes a stochastic multi-scenario construction heuristic, and solves a two-stage stochastic program with recourse that considers a set of disruptions and their recovery solutions to determine delivery time windows. We also report the Phase II computational results. In Section 3.6, we give concluding remarks.

### ***3.2 Problem description***

The MIRPTWA is defined on a finite planning horizon where set  $\mathcal{T}$  contains all time periods. We propose a two-stage stochastic programming model by considering a set  $\mathcal{S}$  of disruption scenarios where scenario 0 has no disruptions and each scenario  $s \in \mathcal{S}/\{0\}$  contains exactly one disruption. The solution under scenario 0 is called the *original plan*. For each disruption scenario, the planning horizon is partitioned into two segments. In the first segment, the original plan is executed; in the second, a *recovery plan* is implemented after period  $l_s$ ,  $s \in \mathcal{S}/\{0\}$  when the disruption is

known. The disruption might not occur until a time after  $l_s$ , for example in the case of an approaching hurricane. In other words, we consider the effect of lead time. In reality, multiple disruptions might occur when an original plan is executed. We approximate the true problem by solving a two-stage stochastic programming model recursively. By using the same planning model at the recovery procedure, it can be viewed as a re-planning stage for potential future disruptions.

We assume there is a set of loading ports denoted by  $\mathcal{J}^{\mathcal{L}}$ . The  $i$ -th port has a constant production rate  $p_{i,t}$  of a single product at period  $t \in \mathcal{T}$  and an initial inventory level  $I_{i,0}$ . The variable  $I_{i,t}^s$  tracks the inventory level at port  $i \in \mathcal{J}^{\mathcal{L}}$  at the end of period  $t \in \mathcal{T}$  under scenario  $s \in \mathcal{S}$ , and it must be between a lower limit  $L_i$  and an upper limit  $U_i$  in each period.

We consider a set of discharging ports denoted by  $\mathcal{J}^{\mathcal{D}}$ . Contractual agreements (made prior to the planning decisions considered here) stipulate that the vendor must deliver at least quantity  $Q_j^{\text{Port}}$  of product at discharging port  $j \in \mathcal{J}^{\mathcal{D}}$  over the entire planning horizon. We use a heterogeneous fleet of ships denoted by  $\mathcal{V} = \{1, 2, \dots, V\}$ , with each ship  $v \in \mathcal{V}$  having some load capacity  $Q_v^{\text{Ship}}$  to deliver the product. For simplicity, we restrict our attention to problems where ships always fully load or discharge at a port, and assume that ships leave a port immediately after they load or discharge the product. The agreements also stipulate that the product should be delivered fairly evenly throughout the planning horizon. Thus, at each discharging port  $j \in \mathcal{J}^{\mathcal{D}}$ , we define a set  $\mathcal{K}_j$  of *time intervals* which corresponds to subsets of consecutive periods, and specify a targeted delivery quantity  $q_{jk}$  for each time interval  $k \in \mathcal{K}_j$ . These targets do not have to be met exactly, but there is a penalty cost  $P^{\mathcal{D}}$  per unit associated with the deviation from the targets. We assume set  $\mathcal{T}_{jk}$  contains all the periods in time interval  $k \in \mathcal{K}_j$  at discharging port  $j \in \mathcal{J}^{\mathcal{D}}$ . The time intervals at a discharging port might overlap with each other. In addition, the berth limit  $B_j$  is the maximum number of ships that can discharge in a given time period at

discharging port  $j \in \mathcal{J}^{\mathcal{D}}$ . We assume that service time (time to load/discharge) is already built into the travel time. Let  $\mathcal{J} = \mathcal{J}^{\mathcal{L}} \cup \mathcal{J}^{\mathcal{D}}$  be the set of all ports.

The planning problem determines a list of time windows at each discharging port. There is an upfront cost  $C_{jw}^{\text{TW}}$  if a  $w$ -day time window is placed at discharging port  $j \in \mathcal{J}^{\mathcal{D}}$ . We assume that the length of each time window is at most  $W$  periods and any two consecutive time windows at a discharging port are separated by at least  $Z$  periods. The time windows are not vessel-specific, but are the same across all the scenarios. A penalty cost  $P^{\text{O}}$  per unit incurs under scenario  $s \in \mathcal{S}/\{0\}$  for each delivery its recovery plan requires outside of the time windows.

We assume that each scenario  $s \in \mathcal{S}/\{0\}$  only includes a single disruption. While multiple disruptions could occur, we do not consider that case in this study. Two types of disruptions are considered in the chapter:

- *Travel disruptions*, such as those caused by a hurricane, affect all routes of ships inbound to a port, resulting in one or multiple day(s) of delay on all routes scheduled to enter the port in a certain time interval.
- *Port disruptions*, such as those caused by a strike or maintenance issue at a specific port, result in one or multiple day(s) of delay for all ships scheduled to load or discharge at the port during a certain time interval.

Mathematically, the two types of disruptions can be defined as follows. Assume  $D(s) = \{(i, t, d)\}$  is the disruption considered in scenario  $s \in \mathcal{S}/\{0\}$ . If it is a travel disruption, then all the ships that are scheduled to arrive at port  $i \in \mathcal{J}$  during periods that are close to  $t \in \mathcal{T}$  are delayed by  $d$  days. If it is a port disruption, then all the ships at the port  $i \in \mathcal{J}$  are not allowed to load or discharge during periods  $t, \dots, t + d - 1$ . Since we assume that ships leave the ports immediately after they load or discharge the product, equivalently, one can interpret the disruptions as (travel disruption): all the ships that are scheduled to arrive at port  $i$  during

periods that are close to  $t \in \mathcal{T}$  are forced to stay at the port by at least  $d$  days before departure, and (port disruption): ships are not allowed to leave port  $i$  during periods  $t, \dots, t + d - 1$ .

The MIRPTWA consists of determining routes for all the ships (including original plan and recovery plan under each scenario  $s \in \mathcal{S}/\{0\}$ ), as well as time windows at all the discharging ports so as to minimize total expected costs of travel, placing delivery time windows, over/under-deliveries, and deliveries outside of the time windows over all the scenarios. We assume equal probability for each scenario  $s \in \mathcal{S}$ .

### 3.3 Mathematical formulations

In this section, we first present a two-stage stochastic mixed-integer programming model as a Full Formulation (FF) for the MIRPTWA and then propose a two-phase heuristic approach to solve the problem.

#### 3.3.1 Full model

The model is constructed on a time-space network. The network has a source node  $n_0$ , a sink node  $n_T$  and a set  $\mathcal{N}$  of regular nodes where each regular node  $n$  is a port-time pair  $(j, t)$ ,  $j \in \mathcal{J}, t \in \mathcal{T}$ . The nodes are shared by all the ships, while each ship has its own travel and waiting arcs in the network. The travel arcs from node  $(j_1, t_1)$  to node  $(j_2, t_2)$  represent travel between ports  $j_1$  and  $j_2$ , and the waiting arcs from node  $(j, t)$  to node  $(j, t + 1)$  represent staying at port  $j$  in both period  $t$  and  $t + 1$ . We assume the travel arcs include the time to load or discharge. We use  $\mathcal{A}$  to denote the set of all arcs, and  $\mathcal{A}^+$  to denote the set of all travel arcs. We assume  $C_a^T$  is the travel cost associated with arc  $a \in \mathcal{A}^+$ . There is no cost for keeping a ship at a port. In addition, the sets of incoming and outgoing travel arcs associated with ship  $v$  and node  $n = (j, t)$  are denoted by  $\mathcal{RS}(j, t, v)^+$  and  $\mathcal{FS}(j, t, v)^+$  respectively, while the sets of incoming and outgoing arcs associated with ship  $v$  and node  $n = (j, t)$  are denoted by  $\mathcal{RS}(j, t, v)$  and  $\mathcal{FS}(j, t, v)$  respectively.

Let  $x_a = 1$  if arc  $a \in \mathcal{A}$  is used, and  $x_a = 0$  otherwise. Let  $y_{jtw} = 1$  if there exists a  $w$ -day time window ( $0 \leq w \leq W$ ) starting from period  $t \in \mathcal{T}$  at discharging port  $j \in \mathcal{J}^D$ , and  $y_{jtw} = 0$  otherwise. The decision variable  $f_{jk}^s$  represents the over-delivery quantity of time interval  $k \in \mathcal{K}_j$  at discharging port  $j \in \mathcal{J}^D$  under scenario  $s \in \mathcal{S}$ . If period  $t \in \mathcal{T}$  at discharging port  $j \in \mathcal{J}^D$  is not covered by any time window, the decision variable  $g_{jt}^s$  represents the delivery quantity under scenario  $s$ ; otherwise,  $g_{jt}^s = 0$ . An arc-flow mixed-integer programming model is given by:

$$(\mathbf{FF}) \quad \min \sum_{a \in A^+} C_a^T x_a^0 + \sum_{j \in \mathcal{J}^D} \sum_{t \in \mathcal{T}} \sum_{w=1}^W C_{jw}^{\text{TW}} y_{jtw} + P^D \sum_{j \in \mathcal{J}^D} \sum_{k \in \mathcal{K}_j} f_{jk}^s + P^O \sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}^D} \sum_{t \in \mathcal{T}} g_{jt}^s \quad (30)$$

$$\text{s.t.} \quad \sum_{a \in \mathcal{FS}(n,v)} x_a^s - \sum_{a \in \mathcal{RS}(n,v)} x_a^s = 0, \quad \forall s \in \mathcal{S}, \forall v \in \mathcal{V}, \forall n \in \mathcal{N}. \quad (31)$$

$$\sum_{a \in \mathcal{FS}(n_0,v)} x_a^s = 1, \quad \forall s \in \mathcal{S}, \forall v \in \mathcal{V}. \quad (32)$$

$$\sum_{a \in \mathcal{RS}(n_T,v)} x_a^s = 1, \quad \forall s \in \mathcal{S}, \forall v \in \mathcal{V}. \quad (33)$$

$$I_{i,t}^s = I_{i,t-1}^s + p_{i,t} - \sum_{v \in \mathcal{V}} Q_v^{\text{Ship}} \sum_{a \in \mathcal{FS}(i,t,v)^+} x_a^s, \quad \forall s \in \mathcal{S}, \forall i \in \mathcal{J}^L, \forall t \in \mathcal{T}. \quad (34)$$

$$L_i \leq I_{i,t}^s \leq U_i, \quad \forall s \in \mathcal{S}, \forall i \in \mathcal{J}^L, \forall t \in \mathcal{T}. \quad (35)$$

$$\sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^s \leq B_j, \quad \forall s \in \mathcal{S}, \forall j \in \mathcal{J}^D, \forall t \in \mathcal{T}. \quad (36)$$

$$\sum_{t \in \mathcal{T}} \sum_{v \in \mathcal{V}} Q_v^{\text{Ship}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^s \geq Q_j^{\text{Port}}, \quad \forall s \in \mathcal{S}, \forall j \in \mathcal{J}^D. \quad (37)$$

$$\sum_{t \in \mathcal{T}_{jk}} \sum_{v \in \mathcal{V}} Q_v^{\text{Ship}} \sum_{a \in FS(j,t,v)^+} x_a^0 - f_{jk} \leq q_{jk}, \quad \forall j \in \mathcal{J}^{\mathcal{D}}, \forall k \in \mathcal{K}_j. \quad (38)$$

$$x_{(i,u,v) \rightarrow (i,u+1,v)}^s \geq \sum_{a \in RS(i,t,v)^+} x_a^s, \quad u = t, \dots, t+d-1, (i, t, d) \in D(s), \forall s \in \mathcal{S}/\{0\}, \forall v \in \mathcal{V}. \quad (39)$$

$$\sum_{u=t}^{t+d-1} \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{FS}(i,t,v)^+} x_a^s = 0, \quad (i, t, d) \in D(s), \forall s \in \mathcal{S}/\{0\}, \forall v \in \mathcal{V}. \quad (39')$$

$$x_a^s = x_a^0, \text{ if arc } a \text{ is a decision to be made before } l_s, \quad \forall s \in \mathcal{S}/\{0\}. \quad (40)$$

$$\sum_{w=1}^W y_{jtw} \leq 1, \quad \forall j \in \mathcal{J}^{\mathcal{D}}, \forall t \in \mathcal{T}. \quad (41)$$

$$y_{jtw} + y_{jt'w'} \leq 1, \quad \forall j \in \mathcal{J}^{\mathcal{D}}, t < t', t + w + Z + 1 \geq t'. \quad (42)$$

$$g_{jt}^s \geq \sum_{v \in \mathcal{V}} Q_v^{\text{Ship}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^s - M \sum_{\tau=t-W+1}^t \sum_{w=t+1-\tau}^W y_{i\tau w}, \quad \forall s \in \mathcal{S}, \forall j \in \mathcal{J}^{\mathcal{D}}, \forall t \in \mathcal{T}. \quad (43)$$

$$x_a^s, y_{jtw} \in \{0, 1\}, f_{jt} \geq 0, g_{jt}^s \geq 0, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \forall j \in \mathcal{J}^{\mathcal{D}}, \forall t \in \mathcal{T}, w = 1, \dots, W. \quad (44)$$

The objective is to minimize total costs of travel, placing delivery time windows, over/under-deliveries in all the considered time intervals, and deliveries outside of the time windows over all the scenarios. Network flow constraints (31) – (33) require each ship to travel from the source to the sink in each scenario. (34) and (35) are

balance constraints of the product at loading ports. (36) are the berth constraints at discharging ports. Constraints (37) and (38) ensure that a minimal quantity of the product is delivered at each discharging port and it is delivered fairly evenly throughout the planning horizon. Constraints (39) and (39') correspond to the travel and port disruptions respectively. Nonanticipativity constraints (40) ensure that the original plan under scenario 0 is executed until period  $l_s$  when the disruption is realized in scenario  $s \in \mathcal{S}/\{0\}$ . Logical constraints (41) and (42) restrict the selection of time windows. Constraints (43) track the deliveries outside of the time windows, where  $M$  is a large number. Constraints (44) are the variable restrictions.

### 3.3.2 Decomposition models for a two-phase heuristic approach

Given the computational complexity, we are not able to consider all possible scenarios or all possible realizations of constraints (39) or (39') in a single model. However, most of the scenarios would not affect the original schedule if they disrupt a route on which there is no ship or a port where no loading/discharging is happening at the time.

As a remedy, we propose a two-phase heuristic approach to solve the problem. In Phase I, we generate routes by incorporating some robustness strategies into a deterministic inventory routing problem without considering disruption scenarios. In Phase II, given the routes obtained in Phase I, we determine delivery time windows by solving a restricted version of Model (FF) by including *heuristic constraints* that are used to fix the ships' visit sequences and a selected set of disruptions that may affect the given routes. When  $\mathcal{S} = \{0\}$ , i.e., when no disruptions are considered, the base model in Phase I is given as:

$$(\mathbf{P1F}) \quad \min \sum_{a \in \mathcal{A}^+} C_a^T x_a^0 + P^D \sum_{j \in \mathcal{J}^D} \sum_{k \in \mathcal{K}_j} f_{jk} \quad (45)$$

$$\text{s.t. (31) -- (38),}$$



$$x_a^0 \in \{0, 1\}, \quad f_{jt} \geq 0, \quad \forall a \in \mathcal{A}, \quad \forall j \in \mathcal{J}^{\mathcal{D}}, \quad \forall t \in \mathcal{T}. \quad (46)$$

Based on (P1F), we introduce two planning strategies to generate robust routes in Phase I, which will be discussed in detail in Section 3.4. The resulting solutions of the Phase I model are routes for each ship that consist of visit sequences as well as loading and discharging times.

Given the solutions to the Phase I model, we extract the information regarding the visit sequences from the solutions and use it as an input in the Phase II model. Although the ships' visit sequences are given, the timing decisions may vary across all the scenarios. Let set  $\mathcal{H}^O$  ( $\mathcal{H}^I$ ) contain all the triples  $(i, t, v)$  if ship  $v$  departs from (arrives at) port  $i$  during time interval  $[t - e, t + e]$  in the Phase I solution, where  $e$  is a small integer. With these restrictions, the Phase II model with the heuristic constraints can be given as:

$$(\mathbf{P2F}) \quad \min \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{t \in \mathcal{T}} \sum_{w=1}^W C_{jw}^{\text{TW}} y_{jtw} + P^{\mathcal{D}} \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{k \in \mathcal{K}_j} f_{jk} + P^O \sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{t \in \mathcal{T}} g_{jt}^s \quad (47)$$

$$\text{s.t. } (31) - (38), (39) \text{ or } (39'), (40) - (44),$$

$$x_a^s = 0, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{FS}(i, t, v)^+, \quad (i, t, v) \notin \mathcal{H}^O, \quad (48)$$

$$x_a^s = 0, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{RS}(i, t, v)^+, \quad (i, t, v) \notin \mathcal{H}^I. \quad (49)$$

Since the visit sequences are fixed in Phase II, we do not include the travel cost in objective function (47). Constraints (48) and (49) ensure that in each scenario, loading and discharging times may be changed from the Phase I solutions even if the visit sequences are given. Algorithm 2 provides an overview of our two-phase heuristic approach.

---

**Algorithm 2:** A Two-phase Heuristic Approach

---

**Phase I** Generate robust routes.

**1.1** Solve a deterministic inventory routing problem with robustness strategies.

- Evenly allocate idle time (see Section 3.4.1).
- Separate deliveries with minimum time requirement (see Section 3.4.2).

**1.2** Fix the visit sequence of each ship

**Phase II** Determine delivery time windows (see Section 3.5).

**2.1** Apply a multi-scenario construction heuristic to obtain initial feasible solutions

**2.2** Solve a two-stage stochastic program with recourse that considers a set of disruptions and their recovery solutions.

---

### 3.3.3 An aggregate model based on ship class

In order to reduce the solution times in (P1F), we use an aggregate model in which ship data are aggregated by ship class so that individual ship schedules are not distinguished in solutions. Such an aggregation technique was used in [59] to obtain a coarse solution in their system model. To apply the aggregate model, we create a variant of the time-space network described in Section 3.3.1, in which the arc set  $\mathcal{A}_{vc}$  for each ship class  $vc \in \mathcal{VC}$  is the union of the arc sets for all the ships in the ship class. Assume there are  $N_{vc}$  ships in the ship class  $vc$ , then we replace constraints (31) – (33) and (46) in (P1F) with the following ones in the aggregate model, which is referred to as **(P1FVC)**.

$$\sum_{a \in \mathcal{FS}(n,vc)} x_a^0 - \sum_{a \in \mathcal{RS}(n,vc)} x_a^0 = 0, \quad \forall vc \in \mathcal{VC}, \forall n \in \mathcal{N}. \quad (31')$$

$$\sum_{a \in \mathcal{FS}(n_0, vc)} x_a^0 = N_{vc}, \quad \forall vc \in \mathcal{VC}. \quad (32')$$

$$\sum_{a \in \mathcal{RS}(n_T, vc)} x_a^0 = N_{vc}, \quad \forall vc \in \mathcal{VC}. \quad (33')$$

$$x_a^0 \in \{0, 1, \dots, N_{vc}\}, \quad f_{jt} \geq 0, \quad \forall a \in \mathcal{A}_{vc}, \quad \forall vc \in \mathcal{VC}, \quad \forall j \in \mathcal{J}^D, \quad \forall t \in \mathcal{T}. \quad (46')$$

A solution produced by (P1FVC) specifies routes for all ship classes, but not for individual ships. To assign a route for each ship, we perform a post-processing step by solving a trivial feasibility problem with constraints (31) – (33) and  $\sum_{a \in \mathcal{A}_{a'}} x_a^0 = \bar{x}_{a'}^0, \quad \forall a' \in \mathcal{A}_{vc}, \forall vc \in \mathcal{VC}$ , where each set  $\mathcal{A}_{a'}$  contains the arcs that are aggregated into the arc  $a'$ , and  $\bar{x}_{a'}^0$  is the optimal value obtained by solving (P1FVC).

### 3.4 Phase I: Robust routing

In this section, we introduce two robustness strategies to generate robust routes in Phase I: *slack reallocation* and *delivery separation*.

#### 3.4.1 Slack (idle time) reallocation

Since a ship might have more time than needed to get from one port to another, there can be some slack in planning solutions. We define *slack days* as the days when a ship is idle at a discharging port. From a robustness point of view, slack days can be regarded as a buffer to protect on-time deliveries. To avoid the cases where some deliveries are over-protected while others are very fragile, we propose a slack reallocation strategy in order to “evenly” allocate the slack among all the deliveries. By using the modeling technique shown in Figure 8, we can track the number of slack days associated with each delivery. In the given example, the normal travel time from loading port A to discharging port B is 3 days and the travel cost is  $c$ . We artificially

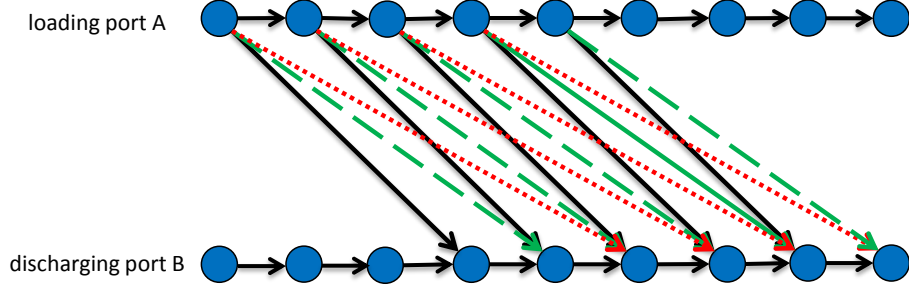


Figure 8: Slack reallocation

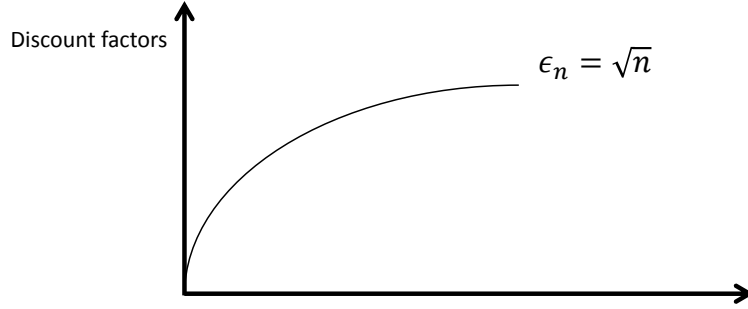


Figure 9: Discount factor for longer travel arcs

include 4-day and 5-day travel arcs from loading to discharging ports on a standard time-space network, and associate them with slightly lower costs  $c - \epsilon_1$  and  $c - \epsilon_2$  respectively where  $0 < \epsilon_1 < \epsilon_2$ . Since using a 4-day travel arc is equivalent to using a normal 3-day travel day (starting from the same period) plus a waiting arc but has a lower cost, the optimization problem would prefer the former one. Therefore, for an optimal solution, we can obtain the number of deliveries that are protected by no slack days, 1 slack day and at least 2 slack days by counting the usage of 3-day (from loading to discharging ports), 4-day and 5-day travel arcs.

More generally, suppose a maximum of  $R$  slack days can give benefit as a buffer. To achieve the goal of “evenly” allocating the slack amongst all the deliveries, we assume that the difference between having a  $r$ -day slack and having a  $(r + 1)$ -day is more significant than the difference between having a  $r'$ -day slack and having a  $(r' + 1)$ -day if  $r < r' \leq R$ . Therefore, the discount factors of using artificial travel arcs are set as a concave function given in Figure 9.

Let set  $\mathcal{A}_r^+$  contain all the artificial travel arcs with an extra  $r$  travel days from loading to discharging ports ( $r = 1, \dots, R$ ) and  $\mathcal{A}_0^+$  contain all the normal travel arcs. We give a discount for ships using artificial slow travel arcs; namely,  $C_{a'}^T = C_a^T - \sqrt{r}$  if  $a \in \mathcal{A}_0^+, a' \in \mathcal{A}_r^+$  and they represent the same port-to-port travel. To incorporate the slack reallocation strategy, we modified the objective function (45) as:

$$\min \sum_{r=0}^R \sum_{a \in \mathcal{A}_r^+} C_a^T x_a^0 + P^D \sum_{j \in \mathcal{J}^D} \sum_{k \in K_j} f_{jk}. \quad (45')$$

### 3.4.2 Separating consecutive deliveries

It is more likely that a disruption could cause significant problems when deliveries are clustered together during a short time frame at a port. Therefore, in order to mitigate the potential effect of a disruption, we separate any two consecutive deliveries at a port by a minimal number of periods. Suppose there are no more than one delivery within any consecutive  $U$  periods at a port, then we call  $U$  a separation parameter and rewrite constraints (36) as

$$\sum_{u=t}^{t+U-1} \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^0 \leq 1, \quad \forall j \in \mathcal{J}^D, \forall t \in \mathcal{T}. \quad (36')$$

### 3.4.3 Phase I computational results

In this section, we show the effects of the two robustness strategies on generating ship routes. Time windows will be placed in Phase II. We expect to see that evenly distributing slack days among the deliveries and setting minimum separation times between consecutive deliveries at a port result in a modest increase in the total travel cost relative to ignoring such robustness strategies altogether. The benefits of coupling these robust routes with judiciously chosen time windows will be shown in Section 3.5.3.

A total of 10 instances are created based on the problem described in Section 3.2. In each instance, we define a 60 periods problem in which there are 2–4 loading ports,

2 – 8 discharging ports, 2 ship groups consisting of 3 – 8 ships. We include 1 extra day, 2 extra days and 3 extra days travel arcs while applying the slack reallocation strategy, namely  $R = 3$ , and we let the separation parameter  $U$  vary from 1 to 4 in the experiments. The integer programs are solved using CPLEX 12.5 with a 4-hour time limit.

Figure 10 shows the average travel cost of the solutions relative to base model. The vertical axis represents the percentage of cost increase based on the basic Phase I model (P1F), which optimistically assumes that there are no disruptions. The horizontal axis gives the separation parameter  $U$  defined in the second robustness strategy. The dotted and solid lines correspond to the models with and without the slack reallocation strategy, respectively. Since any feasible solution to a robust model is feasible to (P1F), we observe in Figure 10 that the average travel cost increases when the robustness strategies are applied. For any given separation parameter, this cost increase is roughly between 0.4% and 0.8% on average. Since the benchmark travel cost used here assumes that no disruptions occur, an average cost increase under 1% seems acceptable.

Figure 11 shows the slack distribution of the best solution found by various models. The horizontal axis represents the average number of deliveries over all the instances with various slack days, and the vertical axis represents different models where SR indicates slack reallocation and  $U$  is the separation parameter. We observe that when the slack reallocation strategy is applied, the number of deliveries that are protected by 1 slack day largely increases while the number of deliveries associated with no slack and more than 2 slack days decreases. Therefore, it achieves our objective of “evenly” allocating the slack amongst all the deliveries.

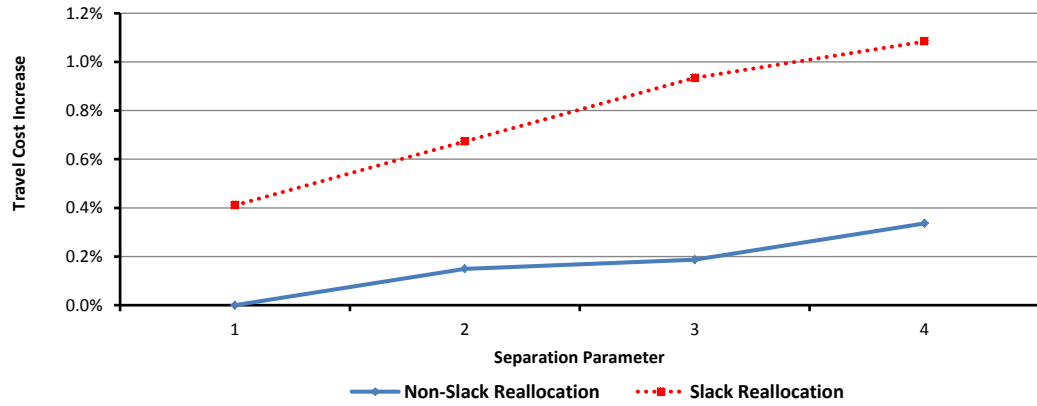


Figure 10: Phase I – Travel cost comparison relative to model without robustness strategy

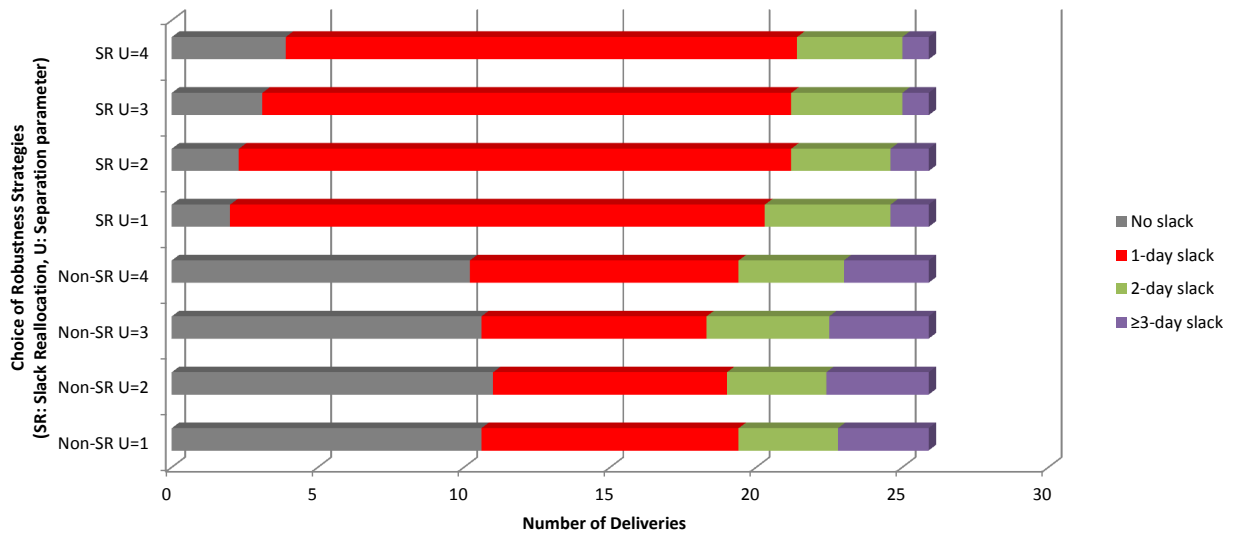


Figure 11: Phase I – Slack day comparison

### **3.5 *Phase II: time window placement***

In Phase I, we obtain routes for ships that consist of both visit sequences and timing for loading and discharging. In Phase II, we allocate time windows for each delivery by solving a restricted version of Model (FF) in which the visit sequences of each ship, obtained in the Phase I solutions, are fixed.

#### **3.5.1 Disruption scenarios**

Since the visit sequence is given for each ship while the loading and discharging times may vary across all the scenarios, constraints (48) and (49) force the value of a large set of decision variables to be 0 in the time-space network. In other words, we largely reduce the number of scenarios that might cause a disruption to the original schedule. To help understand how a disruption might affect travel, loading or discharging of a ship, we illustrate the two types of disruptions in Figure 12. The dashed lines represent original routes, and the solid lines represent recovery routes. In these two examples, a ship is affected due to a travel disruption (in the left one) and a port disruption (in the right one) that both last for 2 periods. As a result, at least two deliveries are delayed due to the cascading effect. Although it shows no difference between these two disruptions in terms of their effects on this particular ship, they may cause different overall problems to the entire schedule. The travel disruption can delay some other inbound travel arcs from L1 to D1 as well that are close in period to the disrupted travel arc shown in the Figure. On the other hand, the port disruption can affect other ships that were prepared to discharge during the disrupted periods.

#### **3.5.2 Construction heuristic**

We propose a stochastic multi-scenario construction heuristic to obtain initial feasible solutions to (P2F). It can be viewed as an extension of the optimization-based local search heuristic that was first introduced in [74]. In their approach, in order to improve an initial feasible solution, they solve restricted optimization models. They



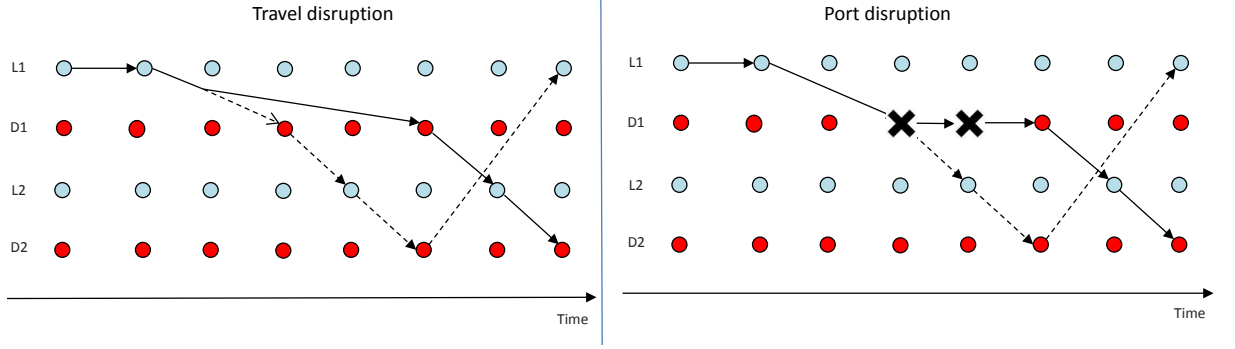


Figure 12: Types of disruptions and their impact on the network

fix the decision variables associated with all but two ships, optimize over those two ships, and repeat this procedure for all ship pairs. [36] adapts this approach as a construction heuristic to generate solutions to instances with 365 time periods. In this chapter, we propose a two-stage stochastic multi-scenario construction heuristic. The heuristic takes as input a set of feasible routes, one for each ship; that solution is obtained by solving a deterministic routing problem such as the one proposed by [80]. In the first stage of the heuristic, we partition the ships into  $VG$  ship groups  $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_{VG}$  based on how closely in time they visit the same port, and for each ship group, we solve a reduced optimization problem that considers only scenario 0 and every other scenario whose disruption may affect one or more ships in  $\mathcal{V}_{vg}$ ,  $vg \in \{1, 2, \dots, VG\}$ . We merge all the individual ship-group solutions together to create an initial solution, which might not be feasible due to the inventory and berth constraints. Therefore, we proceed to a second stage similar to that of [74], but generalized to allow sets of up to  $K$  ships. The second stage emphasizes eliminating feasibility until a feasible solution is obtained; once we have a feasible solution, we continue running improvement iterations in order to improve the quality of solutions. The output of the heuristic is the best solution found, which is then given to the full model as a starting solution for the MIP. An overview of the heuristic is given in Algorithm 2.

---

**Algorithm 3:** Stochastic Multi-scenario Construction Heuristic

---

**Input:** Set of routes for each ship (under no disruptions)

**begin**

**Construction Stage**

1. Partition the ships into  $VG$  ship groups.

1.1 For each ship pair  $v_1, v_2 \in \mathcal{V}$ , calculate  $cl(v_1, v_2)$ , which is the smallest difference in time between a loading/discharging of  $v_1$  and a loading/discharging of  $v_2$  at the same port.

1.2 Partition all the ships into  $VG$  ship groups such that

$\sum_{vg=1}^{VG} \sum_{v_1, v_2 \in \mathcal{V}_{vg}} cl(v_1, v_2)$  is minimized. Each ship group has at least one ship.

2. **for**  $vg = 1, 2, \dots, VG$  **do**

2.1 Create a set  $\mathcal{D}_{vg}$  that contains scenario 0 and the scenarios whose disruption may affect one or more ships in  $\mathcal{V}_{vg}$ .

2.2 Solve a reduced two-stage stochastic MIP over the scenario set  $\mathcal{D}_{vg}$  by only including the decision variables associated with the ships in  $\mathcal{V}_{vg}$ .

2.3 Obtain the solutions for the ships in  $\mathcal{V}_{vg}$  under the scenarios in  $\mathcal{D}_{vg}$ .

3 Merge the solutions.

**for each pair**  $(v, s)$  **where**  $v \in \mathcal{V}_{vg} \subset \mathcal{V}$  **and**  $s \in \mathcal{S}$  **do**

if  $s \in \mathcal{D}_{vg}$  **then**

use the solution obtained in 2.3,

**else**

use the solution under no disruption.

**Improvement Stage**

4. **for**  $k = 1, \dots, K$  **do**

for  $i_1 = 1, \dots, V$  **do**

.....

for  $i_k = i_{k-1} + 1, \dots, V$  **do**

4.1 Fix decision variables associated with each ship

$v, v \notin \mathcal{VF} = \{i_1, \dots, i_k\}$ .

4.2 Solve the restricted two-stage stochastic MIP with the remaining variables.

4.3 Update the current solution.

### 3.5.3 Phase II computational results

In Section 3.4.3, we observed that by incorporating the two robustness strategies in Phase I, there is a modest increase in the total travel cost relative to the basic model which is overly optimistic by assuming no disruptions. In this section, we show the benefits of generating such robust ship routes when we determine delivery time windows. We expect to see that despite a modest increase in the travel cost, applying the robustness strategies enables the vendor to commit relatively small time windows to its customers. Furthermore, compared with the modest increase in generating robust routes, we achieve much more savings by placing small delivery time windows and paying a lower expected recovery cost in case of potential disruptions.

We present the Phase II computational results of the 10 test instances described in Section 3.4.3. We extract the information regarding the visit sequences of ships from the solutions generated in Phase I, and fix them as heuristic constraints in (P2F). Since various planning strategies can lead to different visit sequences, we compare their performance based on the following metrics: travel cost, time-window cost, and total cost of the original plan; and the expected total cost and worst-case-scenario total cost. The number of disruption scenarios considered in (P2F) depends on the number of deliveries in the original plan. Specifically, we include one disruption scenario for each delivery. In our experiments, the number of disruption scenarios varies from 14 to 41 among different instances. For each delivery at a discharging port, we create a disruption scenario with a 2-day delay for all ships on inbound routes to the port. The lead time is assumed to be 7 days, and the maximal length of a time window is 3.

Table 5 shows the comparisons of the solutions generated under two different planning strategies. For each instance, we generate two visit sequences in Phase I, and give them as an input to (P2F). In the first one (VS1), no robustness strategy is used, while in the second one (VS2), we apply the strategy of slack reallocation with

separation parameter  $U = 3$ . The numbers in Table 5 show the percentage in cost change (where a negative value is a cost improvement) of VS2 solutions compared to VS1 solutions. For example, in instance 1, although there was a 0.66% cost increase associated with travel costs, the percentage of cost decrease related to time-window costs was 18.92% and the total scenario 0 cost decreased by 3.17% when using a robust solution over that of ignoring robustness issues. Moreover, the scenario-based expected total cost and the worst-case scenario total cost dropped by 4.12% and 7.11%, respectively, by incorporating the robustness strategies. Since we incorporate the slack reallocation and delivery separation constraints when generating VS2, scenario 0 travel costs cannot decrease. The important observation is that generating routing solutions judiciously with slack reallocation and delivery separation leads to lower time window costs, and decreases the total scenario 0 cost in each of the instances. Furthermore, by applying the two robustness strategies, we have lowered the expected total costs and the worst-case-scenario costs as well, in every instance.

Table 5: Phase II: Cost comparisons for various visit sequences

Ins.	Original plan travel cost	Time-window cost	Original plan total cost	Scenario-based expected total cost	Worst-case scenario total cost
1	0.66%	-18.92%	-3.17%	-4.12%	-7.11%
2	3.82%	-22.81%	-3.27%	-4.22%	-3.10%
3	0.31%	-3.41%	-0.48%	-0.60%	-2.10%
4	1.51%	-9.80%	-0.32%	-0.34%	-0.31%
5	1.41%	-30.16%	-5.80%	-5.78%	-7.05%
6	0.31%	-18.56%	-4.07%	-4.18%	-4.91%
7	0.39%	-2.78%	-0.30%	-0.29%	-0.30%
8	1.14%	-33.33%	-6.74%	-7.68%	-6.90%
9	1.00%	-29.55%	-4.49%	-4.49%	-4.49%
10	0.00%	-12.28%	-1.92%	-2.12%	-2.16%

Figure 13 shows the percentages of the time windows with different sizes when we use the VS2 solution as an input to (P2F) for each instance. We observe that by assuming a 2-day delay for each disruption in our experiments, roughly 60% – 80%

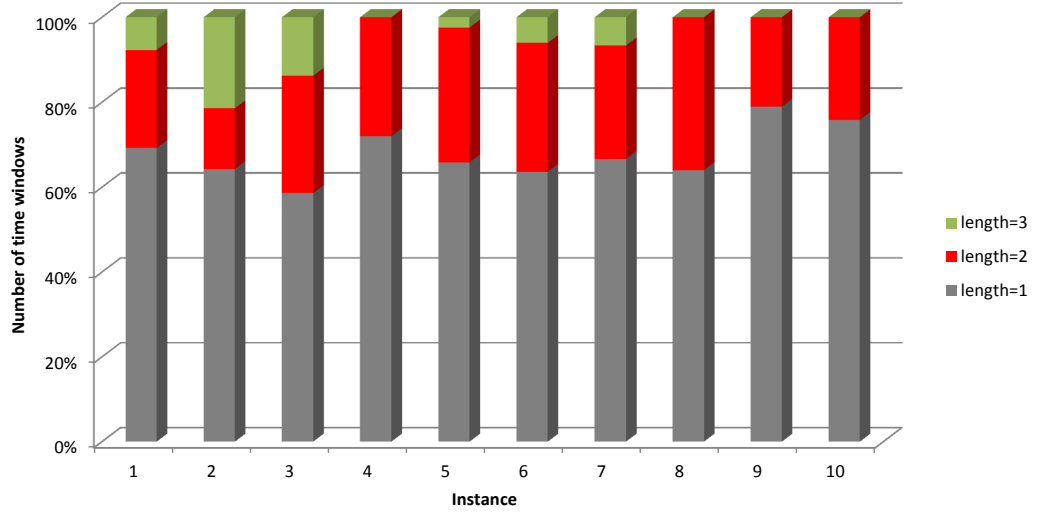


Figure 13: Phase II: Time window distribution

of the time windows contain only 1 day.

We also conducted an experiment to compare our approach with a naive heuristic in which we increase every travel arc by 1 day and assign a 2-day time window to each delivery. In terms of the penalty costs due to the deliveries outside of the time windows, this naive heuristic performs similar to the benchmark strategy in which we use VS1 solutions as an input to (P2F). However, this naive heuristic yields a very high up-front cost of placing time windows which reveals that placing a 2-day time window for each delivery is too conservative. Consequently, our approach in which we use VS2 solutions as an input to (P2F) outperforms this naive heuristic by an even larger margin in terms of the total cost. In summary, this comparison provides further evidence for the need to judiciously place time windows in order to balance costs with robustness.

In Table 6, we report the solution times (in seconds) if solved to optimality, values of the best solutions found by the heuristic (if applied), primal values and dual bounds given by the full model, and optimality gaps. For each instance, we show the number of scenarios (including the scenario with no disruptions) in the second column. A limit of 4 hours is given to the full model. For each instance, there are three rows in

the table, which correspond to the cases where (1) we solve the full model without the heuristic, (2) we use the heuristic with  $K = 1$  before solving the full model, and (3) we use the heuristic with  $K = 2$  before solving the full model, respectively. By using the two-phase stochastic multi-scenario construction heuristic (either with  $K = 1$  or  $K = 2$ ), we achieve a better performance of the full model, and for all the instances, the best solution obtained is at least as good as the best solution found by solving the full model without the construction heuristic. Another important observation is that as we increase the number of iterations at the improvement stage of the heuristic, we improve the quality of solutions in 8 out of the 10 instances; in one of the other two, an optimal solution to the full problem is readily obtained by using the heuristic with  $K = 1$ . However, in terms of the overall performance, it is unclear whether applying one or two improvement iterations of the construction heuristic is better.

### ***3.6 Iterations: updating routes and time-window placement***

In Chapter II, we study a robust MIRPTW with uncertain disruptions, where the delivery time windows are given as an input. A Lagrangian heuristic is introduced to find robust routing solutions under the given time windows. In this Chapter, we study a robust MIRPTW and stochastic travel times, where the length and placement of the time windows are also decision variables. A two-phase approach is proposed in which we generate robust routes in Phase I, and optimize delivery time windows in Phase II. In this section, we conduct computational experiments for an iterative procedure between the heuristic approach proposed in Chapter II and the Phase II model in this Chapter. More specifically, we first try to improve the routes using the Lagrangian heuristic by fixing the delivery time windows determined in Phase II, and then re-optimize the time windows under the new visit sequences. We call a procedure that combines the above two steps Phase III.

Table 6: Phase II: Computational performances of the construction heuristic

Ins.	No. of scenarios	Best solution (CH)	Best solution (Full)	Bound	Gap	Time
1	15	–	30	30	–	315.7
		39	30	30	–	502.07
		31	30	30	–	78.8
2	15	–	92	92	–	77.3
		92	92	92	–	49.8
		92	92	92	–	20.7
3	30	–	132	122.4	7.25%	–
		133	132	132	–	10720.0
		132	132	132	–	5131.8
4	26	–	92	80.4	12.62%	–
		95	92	75.8	17.66%	–
		92	92	92	–	11596.0
5	42	–	419	119.8	71.40%	–
		171	171	134.9	21.11%	–
		171	171	123.7	27.65%	–
6	34	–	138	84.3	38.94%	–
		148	128	111.2	13.09%	–
		133	128	103.4	19.25%	–
7	31	–	103	52.1	49.44%	–
		93	82	67.5	17.72%	–
		91	91	61.30	32.65%	–
8	26	–	58	58	–	8049.6
		111	58	58	–	1592.9
		67	58	58	–	3951.0
9	20	–	31	31	–	1002.8
		33	31	31	–	292.6
		31	31	31	–	260.0
10	30	–	55	55	–	14180.3
		78	55	55	–	3520.7
		63	55	55	–	9591.6

For a given time window, an early delivery is preferred to a late one from the perspective of robustness. The essence of the Lagrangian heuristic is to “prioritize” such preferences among all the time windows by updating the Lagrangian multipliers. Similar as in Chapter II, we set the limit on the maximum number of iterations of Lagrangian heuristic as 10. The integer programs are solved using CPLEX 12.5. The solver is stopped after 3600 seconds. If no integer solution has been found, another 3600 seconds is given, repeating until an integer solution is found. After obtaining new routes, we run the stochastic programming model in Phase II once again but with the updated visit sequences. All the computational parameters remain the same

as in Section 3.3.5.

Table 7 and Table 8 show the percentage in cost change (where a negative value is a cost improvement) of VS1 and VS2 solutions obtained in Phase III compared to in Phase II, respectively. Since the Lagrangian heuristic method improves the robustness of solutions only when time windows are fixed, by re-optimizing the placement of time windows in the second part of Phase III, we do not necessarily achieve a better result in Phase III compared to Phase II. This can be clearly observed in Table 8, where the original plan total cost, the scenario-based expected total cost and the worst-case scenario total cost all get worse (become larger) relative to the results in Phase II in 9 out of the 10. Relatively speaking, the iterative procedure in Phase III performs better for VS1 solutions (Table 7) since no robustness strategies are applied when we generate the routes for VS1 solutions in Phase I. The original plan total cost, the scenario-based expected total cost and the worst-case scenario total cost reduced by more than 2% in 4 out of the 10 instance, while those costs increased by more than 2% for only one instance.

Table 7: Cost changes of VS1 solutions in Phase III compared to in Phase II

Ins.	Original plan travel cost	Time-window cost	Original plan total cost	Scenario-based expected total cost	Worst-case scenario total cost
1	0.00%	-24.32%	-4.76%	-4.51%	-4.57%
2	2.55%	-5.26%	0.47%	0.30%	0.44%
3	0.00%	-13.64%	-2.90%	-2.79%	-2.80%
4	0.38%	21.57%	3.80%	3.70%	3.73%
5	1.41%	-6.35%	-0.36%	-0.37%	-0.53%
6	0.31%	2.06%	0.72%	0.89%	0.70%
7	1.56%	0.00%	1.22%	1.39%	1.19%
8	0.76%	-20.51%	-4.11%	-4.69%	-4.31%
9	1.00%	-25.00%	-3.67%	-3.67%	-3.67%
10	0.00%	8.77%	1.37%	1.37%	1.35%

### 3.7 Summary

In this chapter, we study a robust maritime inventory routing problem with delivery time windows and stochastic travel times, where the length and placement of the time



Table 8: Cost changes of VS2 solutions in Phase III compared to in Phase II

Ins.	Original plan travel cost	Time-window cost	Original plan total cost	Scenario-based expected total cost	Worst-case scenario total cost
1	0.00%	-6.67%	-1.09%	-1.09%	-1.09%
2	0.00%	0.00%	0.00%	0.00%	0.00%
3	0.00%	0.00%	0.00%	0.00%	0.00%
4	0.00%	15.22%	2.22%	2.12%	2.18%
5	-0.93%	22.73%	3.08%	3.10%	4.93%
6	0.62%	-1.27%	0.25%	0.29%	0.25%
7	1.16%	4.29%	1.83%	1.89%	1.80%
8	0.00%	11.54%	1.89%	1.89%	1.85%
9	0.00%	0.00%	0.00%	0.00%	0.00%
10	1.95%	20.00%	4.48%	4.54%	4.70%

windows are also decision variables. We cast the problem as a two-stage stochastic mixed-integer program and propose a two-phase solution approach that considers a sample set of disruptions as well as their recovery solutions. Two planning strategies are proposed to generate robust routes, and a multi-scenario construction heuristic is introduced when we determine the placement of time windows. We also investigate an iterative procedure between updating the routes and re-optimizing the time windows by coupling the Lagrangian heuristic approach proposed in Chapter II.

The problem of determining the length and placement of time windows has largely been overlooked in the vehicle/inventory routing literature, and to the best of our knowledge, this is the first one to consider such a problem as it arises in inventory routing. We believe that time windows play an important role in helping a vendor design and negotiate long-term delivery contracts with its customers. By using an integrated solution procedure that simultaneously considers the routing and time window allocation, we generate committed delivery schedules and routing solutions that are robust against unplanned disruptions and have a low cost of achieving the robustness.

## CHAPTER IV

# ROBUST SOLUTIONS TO A SINGLE-ITEM UNCAPACITATED LOT-SIZING PROBLEM WITH MACHINE BREAKDOWNS

### 4.1 *Introduction*

In this chapter, we study a robust single-item uncapacitated lot-sizing problem (LS-U) with backlogging and random machine breakdowns. The deterministic LS-U is to determine the optimal lot size of a single product at each period, in order to minimize the total production and inventory costs. If unmet demand remains in the system and can be met in a later time, we call the demand backlogged. By assuming that a fixed batch size of the product is produced every time a job is placed, we study a variation of the LS-U, whose objective is to determine the optimal production times on a single machine subject to random breakdowns, so as to minimize the total costs of production, inventory and backlogging under the worst-case scenario.

There are several types of uncertainties that can be encountered in manufacturing systems. One is the possibility of producing defective items that often need rework or sometimes must be discarded. Another common issue is the occurrence of machine breakdowns. Compared to the former issue, machine breakdowns can be more disruptive since they usually affect the production of a large number of items over a long time horizon. Therefore, robust planning solutions are attractive when such a possibility is non-negligible. In this chapter, we consider a robust LS-U problem (RLS-U) faced by a manufacturer who produces a single product on a delicate machine and has a constant demand to fulfill at each period. Despite the simple problem setting, the min-max mixed integer programming formulation of the RLS-U proposed in this

study is intractable. Instead, we derive robust solutions by studying the properties of the worst-case scenarios of disruptions.

#### 4.1.1 Relevant studies

Production planning problems have been extensively studied over the past decades. Early work includes [43] and [78]. For a comprehensive treatment of this subject, we refer to [62].

Compared with the numerous studies on the deterministic lot sizing problems, the number of those that touch the stochastic nature of the problem, especially from a perspective of machine breakdown, is relatively small. A common assumption that is usually made is that we operate on a reliable machine. However, in practice, machine breakdowns can have an impact throughout the manufacturing process, and thus affect the optimal replenishment policy including when and how much to produce. Depending on the setting of the problems, they can be classified along a number of dimensions such as lost sales vs. backorders, decision variables (order quantity vs. reorder time), distributions of machine failure, repair time and demand. Given the complexity of the problems with random machine disruptions, closed-form solutions or proven optimal policies are usually not available. Instead, many studies compare some classical inventory policies and optimize the parameters of these policies under stochastic settings.

[38] proposes two production control policies to deal with machine breakdowns. One assumes that the production is not resumed after a breakdown, while the on-hand inventory is depleted before a new cycle is initiated. The other policy assumes that production is immediately resumed after a breakdown if the on-hand inventory is below a certain threshold level. Optimal lot sizes are determined when repair time is negligible and when it follows an exponential distribution. This work is extended in [37] in which they assume that a certain fraction of the items produced is diverted

into safety stock. Demand is lost and the safety stock is depleted when the machine is under repair. [1] considers an Economic Production Quantity (EPQ) model with Poisson machine failures and lost sales. Repair times are assumed to be a constant or follow an exponential distribution. [29] and [30] provide bounds to the optimal lot sizes of the models proposed in [38], and show that the long-run average cost function per unit of time for the case of exponential failures is unimodal. [18] develops a unified framework in which preventive maintenance and safety stocks are jointly considered as two strategies that protect against an unreliable machine. Cases with deterministic and exponential repair times are studied in the paper. [53] considers the impact of random machine breakdowns on the classical EPQ model for a product subject to exponential decay and under a no-resumption inventory control policy. The time-to-breakdown is a random variable following an exponential distribution and the repair time is fixed. The objective of the paper is to determine optimal production uptimes that minimize the expected total cost per unit time. [20], [21] and [22] study an EPQ model with scrap, rework and stochastic machine breakdowns, and propose an algorithm to determine optimal run times. Other relevant studies on stochastic machine breakdowns can be found in [2], [17], [35], [47], [48], [54], [55], [56], [61] and [63].

In the studies listed above, it is always assumed that times between machine failures and repair follow a given distribution such as exponential or uniform, or is negligible. The main difference between our study and those in the literature is that instead of making some distribution assumptions, we propose a robust optimization approach after specifying a maximum number of periods of machine breakdown. The production quantity is given as a constant every time a production job is initiated. The objective is to determine the production times so as to minimize the total costs under the worst-case scenario.

### 4.1.2 Contributions

The major contributions of this work are twofold:

1. To the best of our knowledge, this is the first study that considers a RLS-U with random machine disruptions whose objective is to minimize the total worst-case costs. We study the robust optimization problem in a periodic-review model that considers the costs of production, inventory and backlogging.
2. By identifying the solution characteristics of the worst-case disruptions, we demonstrate that a stationary production plan (to be defined in Section 4.3) must be optimal, and show how to efficiently find the optimal plan and its worst-case scenario.

The remainder of this chapter is organized as follows. In Section 4.2, we give a problem description, discuss budgets of uncertainty and provide a min-max mixed-integer programming formulation. In Section 4.3, we show the characteristics of the worst-case disruptions for stationary production plans, and identify the optimal solutions to the robust model. Finally, we give numerical results in Section 4.4.

## 4.2 *Problem description and mathematical formulation*

### 4.2.1 Problem description and budget of uncertainty

The problem is defined on a finite planning horizon where set  $\mathcal{T} = \{1, 2, \dots, T\}$  contains all time periods. We use a single machine to produce a product. The machine can process multiple jobs in parallel. Every time period in which we initiate a job, a fixed batch size  $V$  of the product is produced with zero lead time (i.e., production is completed within that time period). There is a constant demand  $r$  at each period. It is satisfied (or partially satisfied) when inventory is available, and the unsatisfied demand is backlogged. Variables  $I_t^+$  and  $I_t^-$  track the inventory level and stockout level at period  $t$ ,  $t \in \mathcal{T}$  respectively. Inventory can be held from period to

period with no loss or spoilage. The initial inventory level is  $I_0^+$ . We assume that schedules created by the production scheduler will never lead to stockouts except if the machine breaks down.

Due to potential machine disruptions, it might fail to work (off) for certain periods. If a job is scheduled at a period in which the machine is broken, it will be held until the machine starts to work again (e.g., it is repaired or replaced). When we schedule the jobs for the upcoming planning horizon, we have no information regarding whether the machine will be on or off in each period. It is only assumed that the machine will be off in no more than  $u$  periods, i.e.,

$$\sum_{t=1}^T \delta_t \leq u, \quad (50)$$

where the binary variable  $\delta_t$  is 1 if the machine is off at period  $t$  and 0 otherwise.

To illustrate how machine breakdowns can affect a production plan, we show an example below with initial inventory level  $I_0^+ = 110$ , demand per period  $r = 30$  and batch size  $V = 130$ :

Period	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Batch scheduled	--	<i>N</i>	<i>N</i>	<i>Y</i>	<i>N</i>	<i>N</i>	<i>Y</i>	<i>N</i>	<i>N</i>	<i>N</i>	<i>N</i>	<i>Y</i>	<i>N</i>	<i>N</i>
Machine status	on	on	off	off	off	off	off	on	on	on	on	off	on	off
# of batches produced	--	0	0	0	0	0	0	2	0	0	0	0	1	0
Actual inventory	110	80	50	20	0	0	0	160	130	100	70	40	140	110
Stockout level	0	0	0	0	10	40	70	0	0	0	0	0	0	0

In this example, batch jobs are scheduled at periods 3, 6 and 11, and machine disruptions occur at periods 2, 3, 4, 5, 6, 11 and 13. It is obvious that if the machine is off at a period in which no job is scheduled or held, it has no effect on the plan. However, since the machine is off for 5 consecutive periods starting from period 2, the jobs that are scheduled at periods 3 and 6 are not produced until period 7.

#### 4.2.2 A mixed integer programming formulation

Suppose  $x_i$  is a binary decision variable which is 1 if there is a job scheduled at period  $i$  and 0 otherwise,  $i \in \mathcal{T}$ . Let binary variable  $y_{ij}$  be 1 if a job that is scheduled at

period  $i$  is produced at period  $j$  and 0 otherwise,  $i \in \mathcal{T}, j \in \{i, i+1, \dots, i+u\}$  (Since the machine will be off in no more than  $u$  periods, there is no need for  $j$  to be greater than  $i+u$ ). Since a job that is scheduled at a period in  $\mathcal{T}$  could be delayed until period  $T+1$ , in order to avoid this end effect in our model, we assume that the machine always works properly at period  $T$  so that all the scheduled jobs can be completed in the planning horizon. (Or, equivalently, we can allow the machine to fail at period  $T$  but consider the problem in an extended horizon with  $\hat{\mathcal{T}} = \mathcal{T} \cup \{T+1\}$ .) This assumption does not affect the generality of the model since we can always use the model by rolling forward the planning horizon. Let  $\mathcal{S}(i) = \mathcal{T} \cap \{i, i+1, \dots, i+u\}$ . Let  $c_f$  be the cost of producing a batch of size  $V$ ,  $c_h$  be the cost of holding inventory per unit per period, and let  $c_s$  be the cost of a stockout per unit per period ( $c_s > c_h$ ). The robust optimization problem described in Section 4.2.1 can be formulated as a mixed integer program, which we refer to as **(RO)**:

$$\textbf{(RO):} \quad \min_x \max_{\delta} \sum_{i=1}^T \sum_{j \in \mathcal{S}(i)} c_f y_{ij} + \sum_{i=1}^T (c_h I_i^+ + c_s I_i^-) \quad (51)$$

$$\text{s.t.} \quad I_0^+ - ir + V \sum_{k=1}^i x_k \geq 0 \quad \forall i \in \mathcal{T}, \quad (52)$$

$$\sum_{i=1}^T \delta_i \leq u, \quad (53)$$

$$\sum_{j \in \mathcal{S}(i)} y_{ij} = x_i, \quad \forall i \in \mathcal{T}, \quad (54)$$

$$y_{ij} + \delta_j \leq 1, \quad \forall i \in \mathcal{T}, \forall j \in \mathcal{S}(i), \quad (55)$$

$$\sum_{k=i}^j y_{ik} \geq x_i - \delta_j, \quad \forall i \in \mathcal{T}, \forall j \in \mathcal{S}(i), \quad (56)$$

$$I_j = I_{j-1} - r + V \sum_{i=j-u}^j y_{ij}, \quad \forall j \in \mathcal{T}, \quad (57)$$

$$I_j = I_j^+ - I_j^-, \quad \forall j \in \mathcal{T}, \quad (58)$$

$$I_i^+ \geq 0, \quad I_i^- \geq 0, \quad x_i, y_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{T}, \quad \forall j \in \mathcal{S}(i). \quad (59)$$

Objective function (51) minimizes the total costs of production, inventory holding and stockout under the worst-case scenario. Constraints (52) stipulate that we never schedule stockout. Constraints (53) indicate that the machine fails to work in no more than  $u$  periods. Constraints (54) hold since a scheduled job can be postponed for at most  $u$  periods. (55) and (56) are logical constraints indicating that a scheduled or backlogged job is immediately released if the machine is on. Constraints (57) and (58) track the inventory flow, and give the storage and stockout units at each period. Constraints (59) are variable restrictions.

By assuming  $u = 0$ , we have the special case where there are no machine disruptions. Since scheduling stockout is not allowed, the obvious optimal solution to the problem when  $u = 0$  is

$$x_i^* = \begin{cases} 1 & \text{if } 0 \leq I_{i-1}^+ < r, \\ 0 & \text{if } I_{i-1}^+ \geq r. \end{cases}$$

Namely, a job is scheduled at the next period once the current inventory level is below the demand.

### 4.3 *Solution approach*

Since there are a finite number of scenarios, for any given production plan, there exists at least one worst-case scenario in terms of the actual cost incurred. Although the robust optimization problem can be formulated in the previous section as a MIP, it is not computationally tractable. Therefore, to explicitly solve the problem, we



explore the properties of the worst-case scenarios for some given production plans. Suppose  $V = qr$  where  $q$  is an integer number. We call a production plan *stationary* if a job is scheduled every  $q$  periods in the plan. In this section, we answer the following question: for a given uncertainty budget and a stationary production plan, at which periods should the machine fail to work to maximize the total actual cost? Once we have characterized worst-case scenarios for stationary production plans, we demonstrate that a stationary production plan must be optimal, and show how to efficiently find the optimal plan and its worst-case scenario.

### 4.3.1 Notation

We first introduce some notation for ease of expression later. Let  $p = (i_1, i_2, \dots, i_m)$  be a plan in which there is a job scheduled at periods  $i_1, i_2, \dots, i_m \in \mathcal{T}$  respectively, and  $a = (j_1, w_1, j_2, w_2, \dots, j_k, w_k)$  be a scenario in which the machine is off for  $w_1$  consecutive periods starting from  $j_1$ ,  $w_2$  consecutive periods starting from  $j_2$ , etc. In set  $\mathcal{A}_p^u$ , we collect all the scenarios satisfying

1.  $\sum_{l=1}^k w_l \leq u$ ,
2.  $\{j_1, j_2, \dots, j_k\} \subseteq \{i_1, i_2, \dots, i_m\}$ ,
3.  $j_\tau < j_{\tau+1}$ ,  $0 < w_\tau < j_{\tau+1} - j_\tau$ ,  $\tau = 1, 2, \dots, k-1$ .

Condition 1 is the budget of uncertainty. Since a machine disruption has no effect on the plan  $p = (i_1, i_2, \dots, i_m)$  until it gets to a period in set  $\{i_1, i_2, \dots, i_m\}$ , we can restrict our attention to the disruptions with condition 2 without loss of generality. Condition 3 ensures the uniqueness of representation.

Assume  $I_i^+(p, a)$  is the inventory level at period  $i$  under original plan  $p$  and disruption scenario  $a$ . Suppose  $\Phi_{i_\tau}(p, a)$  and  $\Lambda_{i_\tau}(p, a)$  are the additional inventory and stockout costs incurred from period  $i_\tau$  to period  $(i_{\tau+1} - 1)$  under original plan  $p$  and disruption scenario  $a$ , respectively; let  $\Delta_{i_\tau}(p, a) = \Phi_{i_\tau}(p, a) + \Lambda_{i_\tau}(p, a)$  be the total

additional cost incurred from period  $i_\tau$  to period  $(i_{\tau+1} - 1)$ . Suppose  $\Phi_T(p, a)$  and  $\Lambda_T(p, a)$  are the additional inventory and stockout costs incurred over the entire planning horizon, respectively; let  $\Delta_T(p, a) = \Phi_T(p, a) + \Lambda_T(p, a)$  be the total additional cost incurred over the entire planning horizon. The optimal solution to the inner maximization problem of (RO) is  $a_p^*$ ; namely,  $\Delta_T(p, a_p^*) = \max_{a \in \mathcal{A}_p^u} \Delta_T(p, a)$ . Without causing ambiguity, we sometimes omit  $p$  in the above notation for a concise representation. Suppose  $\Pi(p, a)$  is the total actual cost under original production plan  $p$  and disruption scenario  $a$ , and  $\Pi^*(p) = \max_{a \in \mathcal{A}_p^u} \Pi(p, a)$  is the total actual cost under  $p$  and its worst-case scenario.

### 4.3.2 Properties of worst-case scenarios

In this section, we focus on a subset  $\mathcal{X}$  of all the feasible production plans, where  $\mathcal{X} = \{(i_1, i_2, \dots, i_m) : i_{\tau+1} - i_\tau = q, \tau = 1, 2, \dots, m-1\}$  contains all the stationary production plans. We call each time interval  $[i_\tau, i_{\tau+1} - 1]$  a *production cycle*.

Lemma 1 shows that for a scenario with multiple consecutive disruptions, the additional inventory and stockout costs incurred have an additive property.

**Lemma 1** *For any production plan  $p$  and uncertainty budget  $u$ , suppose  $a = (j_1, w_1, j_2, w_2, \dots, j_k, w_k) \in \mathcal{A}_p^u$  and  $a_\tau = (j_\tau, w_\tau)$ ,  $\tau = 1, 2, \dots, k$ , then  $\Phi_T(p, a) = \sum_{\tau=1}^k \Phi_T(p, a_\tau)$ ,  $\Lambda_T(p, a) = \sum_{\tau=1}^k \Lambda_T(p, a_\tau)$ .*

*Proof:* Since demand is backlogged and  $w_\tau < j_{\tau+1} - j_\tau$ , the inventory level at  $j_{\tau+1} - 1$  resumes to the scheduled level under disruption  $a$ ,  $\tau = 1, 2, \dots, k-1$ . Therefore, we have  $\Phi_T(p, a) = \sum_{\tau=1}^k \Phi_T(p, a_\tau)$ . Similarly, we have  $\Lambda_T(p, a) = \sum_{\tau=1}^k \Lambda_T(p, a_\tau)$ .  $\square$

Proposition 3 shows that the worst-case scenario has either one consecutive disruption or no disruptions.

**Proposition 3** Suppose  $V = qr$  where  $q$  is an integer number, and  $u$  is the uncertainty budget. For any stationary production plan  $p = (i_1, i_2, \dots, i_m) \in \mathcal{X}$ , the worst-case scenario  $a_p^* \in \{(s, w), s = i_1, i_2, \dots, i_m, 0 \leq w \leq u\}$ .

*Proof:* We denote the inventory level at period  $i_1 - 1$  as  $s_0$ . Given the periodicity, the inventory level at period  $i_\tau - 1$  is  $s_0$ ,  $\tau = 2, \dots, m$ .  $s_0$  is the *base-stock level* of the stationary production plan  $p$ . We first prove that a scenario that has two consecutive disruptions  $a = (i_1, w_1, i_2, w_2)$  is not the worst-case scenario by showing that the disruption  $a' = (i_1, w_1 + w_2)$  will result in an actual cost at least as high as the disruption  $a$ . This claim is established under the following three cases.

**Case 1:**  $0 \leq s_0 < r$ .

Table 9 and Table 10 give the saved inventory amounts and stockout levels from period  $i_1$  to period  $(i_1 + w_1 + w_2 - 1)$  due to the disruption scenario  $a'$  if  $w_1 + w_2 \leq q$  and  $w_1 + w_2 > q$ , respectively. Let  $(w_1 + w_2) \equiv o \pmod{q}$ , where  $0 \leq o < q$ . Since  $s_0 < r$ , the system has stockout starting from period  $i_1$ , and the actual inventory level for all the periods from  $i_1$  to  $(i_1 + w_1 + w_2 - 1)$  are 0.

Table 9: Saved inventory amounts and stockout levels (case 1,  $w_1 + w_2 \leq q$ )

Period	Inventory saved	Stockout
$i_1$	$s_0 + V - r$	$r - s_0$
$i_1 + 1$	$s_0 + V - 2r$	$2r - s_0$
$\vdots$	$\vdots$	$\vdots$
$i_1 + w_1 + w_2 - 1$	$s_0 + V - (w_1 + w_2)r$	$(w_1 + w_2)r - s_0$

Since the stockout amount is increasing starting from the first affected period, a consecutive  $(w_1 + w_2)$ -period disruption results in a higher stockout cost than two consecutive disruptions with a length of  $w_1$  and  $w_2$ , respectively, i.e.,  $\Lambda_T(p, a') > \Lambda_T(p, a) = \Lambda_{i_1}(p, a) + \Lambda_{i_2}(p, a)$ . On the other hand, the inventory cost saved due to a disruption is decreasing in a production cycle. Combined with the fact that every consecutive disruption starts from the beginning of a production cycle ( $i_1$  or  $i_2$  in

Table 10: Saved inventory amounts and stockout levels (case 1,  $w_1 + w_2 > q$ )

Period	Inventory saved	Stockout
$i_1$	$s_0 + V - r$	$r - s_0$
$i_1 + 1$	$s_0 + V - 2r$	$2r - s_0$
$\vdots$	$\vdots$	$\vdots$
$i_2 - 1$	$s_0$	$qr - s_0$
$i_2$	$s_0 + V - r$	$(q + 1)r - s_0$
$\vdots$	$\vdots$	$\vdots$
$i_1 + w_1 + w_2 - 1$	$s_0 + V - or$	$(w_1 + w_2)r - s_0$

this case), a consecutive  $(w_1 + w_2)$ -period disruption saves less inventory costs than two consecutive disruptions with a length of  $w_1$  and  $w_2$ , respectively, i.e.,  $\Phi_T(p, a') > \Phi_T(p, a) = \Phi_{i_1}(p, a) + \Phi_{i_2}(p, a)$ . Therefore, we conclude that  $\Delta_T(p, a') > \Delta_T(p, a)$ .

**Case 2:  $r \leq s_0 < V$ .**

Suppose  $mr \leq s_0 < (m + 1)r < V$  where  $m$  is a positive integer. Since a disruption always lowers the inventory cost, in order to ensure that  $\Delta_{i_1}(p, a) > 0$  and  $\Delta_{i_2}(p, a) > 0$ , we must have  $w_1 > m$  and  $w_2 > m$ . In other words, both consecutive disruptions result in stockouts. Table 11 and Table 12 give the planned inventory levels, actual inventory levels, saved inventory amounts and stockout levels from period  $i_1$  to period  $(i_1 + w_1 + w_2 - 1)$  due to the disruption scenario  $a'$  if  $w_1 + w_2 \leq q$  and  $w_1 + w_2 > q$ , respectively. Again, let  $(w_1 + w_2) \equiv o \pmod{q}$ , where  $0 \leq o < q$ .

Table 11: Inventory and stockout status (case 2,  $w_1 + w_2 \leq q$ )

Period	Inventory planned	Actual inventory	Inventory saved	Stockout
$i_1$	$s_0 + V - r$	$s_0 - r$	$V$	$0$
$i_1 + 1$	$s_0 + V - 2r$	$s_0 - 2r$	$V$	$0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i_1 + m - 1$	$s_0 + V - mr$	$s_0 - mr$	$V$	$0$
$i_1 + m$	$s_0 + V - (m + 1)r$	$0$	$s_0 + V - (m + 1)r$	$(m + 1)r - s_0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i_1 + w_1 + w_2 - 1$	$s_0 + V - (w_1 + w_2)r$	$0$	$s_0 + V - (w_1 + w_2)r$	$(w_1 + w_2)r - s_0$

The saved inventory cost is non-increasing in every production cycle, and periodic after the second production cycle. Since  $w_1 > m$  and  $w_2 > m$ , by merging the

Table 12: Inventory and stockout status (case 2,  $w_1 + w_2 > q$ )

Period	Inventory planned	Actual inventory	Inventory saved	Stockout
$i_1$	$s_0 + V - r$	$s_0 - r$	$V$	0
$i_1 + 1$	$s_0 + V - 2r$	$s_0 - 2r$	$V$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i_1 + m - 1$	$s_0 + V - mr$	$s_0 - mr$	$V$	0
$i_1 + m$	$s_0 + V - (m + 1)r$	0	$s_0 + V - (m + 1)r$	$(m + 1)r - s_0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i_2 - 1$	$s_0$	0	$s_0$	$qr - s_0$
$i_2$	$s_0 + V - r$	0	$s_0 + V - r$	$(q + 1)r - s_0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i_1 + w_1 + w_2 - 1$	$s_0 + V - or$	0	$s_0 + V - or$	$(w_1 + w_2)r - s_0$

two consecutive disruptions in scenario  $a$  into a consecutive one in scenario  $a'$ , the difference of the saved inventory costs between scenarios  $a$  and  $a'$  is bounded by the following inequality:

$$\Phi_T(a) \leq \Phi_T(a') + c_h \sum_{l=1}^m (s_0 - lr) = \Phi_T(a') + c_h (ms_0 - \frac{m(m+1)}{2}r). \quad (60)$$

On the other hand, the stockout amount is non-decreasing starting from the first affected period. Since  $w_1 > m$  and  $w_2 > m$ , the difference of the stockout costs between scenarios  $a$  and  $a'$  is bounded by the following inequality:

$$\Lambda_T(a) \leq \Lambda_T(a') - c_s \sum_{l=m+1}^{2m+1} (lr - s_0) = \Lambda_T(a') - c_s [\frac{(3m+2)(m+1)}{2}r - (m+1)s_0]. \quad (61)$$

By combining the fact that

$$\begin{aligned}
& c_s [\frac{(3m+2)(m+1)}{2}r - (m+1)s_0] - c_h [ms_0 - \frac{m(m+1)}{2}r] \\
& \geq c_h [\frac{(3m+2)(m+1)}{2}r - (m+1)s_0 - ms_0 + \frac{m(m+1)}{2}r] \\
& = c_h (2m^2r + 3mr + r - s_0 - 2ms_0) \\
& > c_h (2m^2r + 3mr + r - mr - r - 2m^2r - 2mr) \\
& = 0,
\end{aligned}$$

we conclude that  $\Delta_T(a) = \Phi_T(a) + \Lambda_T(a) < \Phi_T(a') + \Lambda_T(a') = \Delta_T(a')$ , i.e., the total inventory and stockout costs incurred under scenario  $a'$  is larger than those incurred under scenario  $a$ .

**Case 3:**  $s_0 \geq V$ .

Suppose  $kV \leq mr \leq s_0 < (m+1)r \leq (k+1)V$  where  $k, m$  are positive integers. There are multiple jobs scheduled before stockout. Since  $V = qr$ ,  $kq \leq m < kq + q$ . Let  $m = kq + k'$ . The system has stockout starting from period  $(m+1)$  in the  $(k+1)$ -th production cycle. Since  $w_1 > m, w_2 > m$  and the stockout amount is non-decreasing starting from the first affected period, the difference of the stockout costs between scenarios  $a$  and  $a'$  can still be bounded by inequality (61).

The saved inventory is  $lV$  for every period in the  $l$ -th production cycle,  $l = 1, 2, \dots, k$ . Therefore, the difference of the saved inventory cost between  $a$  and  $a'$  is bounded by the following inequality:

$$\Phi_T(a) \leq \Phi_T(a') + c_h \left( \sum_{\tau=0}^{k-1} \sum_{l=1}^q (s_0 - \tau V - lr) + \sum_{l=1}^{k'} (s_0 - kV - lr) \right) \leq \Phi_T(a') + c_h \sum_{l=1}^m (s_0 - lr),$$

which means (60) holds as well. Therefore, we have  $\Delta_T(a) = \Phi_T(a) + \Lambda_T(a) < \Phi_T(a') + \Lambda_T(a') = \Delta_T(a')$  given the same reasoning shown in Case 2.

By combining the above three cases, we conclude that the total inventory and stockout costs incurred under disruption  $a' = (i_1, w_1 + w_2)$  is larger than those incurred under disruption  $a = (i_1, w_1, i_2, w_2)$ . For any scenario that consists of more than two consecutive disruption segments, we can show that it is not the worst scenario by repeatedly applying the above conclusion and Lemma 1. Therefore, we have established Proposition 3.  $\square$

Furthermore, Proposition 4 shows that the worst-case scenario is either a single  $u$ -period consecutive disruption or has no disruptions.

**Proposition 4** *Suppose  $V = qr$  where  $q$  is an integer number, and  $u$  is the uncertainty budget. For any stationary production plan  $p = (i_1, i_2, \dots, i_m) \in \mathcal{X}$ , the worst-case scenario  $a_p^* \in \{(s, 0), (s, u), s = i_1, i_2, \dots, i_m\}$ .*

*Proof:* Proposition 3 states that the worst-case scenario  $a_p^* \in \{(s, w), s = i_1, i_2, \dots, i_m, 0 \leq w \leq u\}$ . Assume  $L = \{w : 1 \leq w \leq u, \Delta_T(i_1, w) > 0\}$  is the set of lengths of consecutive disruptions that result in increased total costs. If  $L = \emptyset$ , then  $a_p^* = (i_1, 0)$ . The worst-case scenario has no disruptions; hence, the proposition holds.

If  $L \neq \emptyset$ , let  $l = \min\{w : 1 \leq w \leq u, \Delta_T(i_1, w) > 0\}$  be the shortest length of disruption that results in an increased total cost. Suppose  $l = ql_D + l_R$ , where  $l_D, l_R$  are nonnegative integers and  $0 \leq l_R < q$ . As in Proposition 3, we assume  $s_0$  is the base-stock level and establish Proposition 4 under three cases.

**Case 1:  $0 \leq s_0 < r$ .**

Since the stockout amount is increasing starting from the first affected time period, and the inventory cost saved due to a disruption is decreasing in a production cycle, we know that by extending a disruption until the next production cycle, a larger actual cost incurs. Thus, if  $l_R > 0$ , then

$$\Delta_T(i_1, w + 1) > \Delta_T(i_1, w) \quad , w = l, l + 1, \dots, (l_D + 1)q - 1. \quad (62)$$

Now, we will show that a larger actual cost incurs even when we extend the disruption across a new production cycle, i.e.  $\Delta_T(i_1, (l_D + 1)q + 1) > \Delta_T(i_1, (l_D + 1)q)$ . It is equivalent to the case that at period  $(l_D + 1)q + 1$ , the stockout cost incurred is more than the inventory cost saved due to the disruption, which is

$$c_s[(l_D + 1)q + 1)r - s_0] > c_h(s_0 + V - r).$$

Since  $\Delta_T(i_1, (l_D + 1)q) > 0$ , and  $\Delta_T(i_1, l_D q) \leq 0$  according to the definition of  $l$ , we have

$$\begin{aligned} & \Delta_T(i_1, (l_D + 1)q) - \Delta_T(i_1, l_D q) \\ &= \Lambda_T(i_1, (l_D + 1)q) - \Lambda_T(i_1, l_D q) + \Phi_T(i_1, (l_D + 1)q) - \Phi_T(i_1, l_D q) \\ &= c_s \sum_{\tau=l_D q+1}^{l_D q+q} (\tau r - s_0) - c_h \sum_{\tau=1}^q (s_0 + V - \tau r) \\ &= c_s \left[ \frac{(2l_D q + q + 1)qr}{2} - qs_0 \right] - c_h \left[ qs_0 + qV - \frac{(q+1)q}{2}r \right] \\ &> 0. \end{aligned}$$

Hence, by applying the above result in the first inequality below, we obtain

$$\begin{aligned}
& c_s[((l_D + 1)q + 1)r - s_0] - c_h(s_0 + V - r) \\
= & c_s[\frac{(2l_D q + q + 1)r}{2} - s_0] + c_s \frac{(q+1)r}{2} - c_h(s_0 + V - r) \\
> & c_h[s_0 + V - \frac{(q+1)r}{2}] + c_h \frac{(q+1)r}{2} - c_h(s_0 + V - r) \\
= & c_h r \\
> & 0.
\end{aligned}$$

Therefore,

$$\Delta_T(i_1, (l_D + 1)q + 1) > \Delta_T(i_1, (l_D + 1)q). \quad (63)$$

By repeatedly using (62) and (63) as we keep adding  $w$  by 1 every time, we conclude that  $a_p^* = (i_1, u)$ .

**Case 2:  $r \leq s_0 < V$ .**

Suppose  $mr \leq s_0 < (m + 1)r \leq V$  where  $m$  is a positive integer. Since the stockout amount is non-decreasing starting from the first affected time period, and the inventory cost saved due to a disruption is non-increasing in a production cycle, (62) still follows when  $l_R > 0$ . When  $l_D > 0$ , by the same reasoning given in Case 1, (63) still follows. When  $l_D = 0$ , since  $\Delta_T(i_1, q) > 0$ , we have

$$\begin{aligned}
\Delta_T(i_1, q) &= \Lambda_T(i_1, q) + \Phi_T(i_1, q) \\
&= c_s \sum_{\tau=m+1}^q (\tau r - s_0) - c_h[mV + \sum_{\tau=m+1}^q (s_0 + V - \tau r)] \\
&= c_s[\frac{(m+q+1)(q-m)r}{2} - (q-m)s_0] - c_h[mV + (q-m)(s_0 + V) - \frac{(m+q+1)(q-m)r}{2}] \\
&> 0.
\end{aligned}$$

Hence, by applying the above result in the first inequality below, we obtain

$$\begin{aligned}
& (q - m)[\Delta_T(i_1, q + 1) - \Delta_T(i_1, q)] \\
= & c_s[(q + 1)(q - m)r - (q - m)s_0] - c_h(q - m)(s_0 + V - r) \\
= & c_s[\frac{(m+q+1)(q-m)r}{2} - (q - m)s_0] + c_s \frac{(q-m)(q-m+1)r}{2} - c_h(q - m)(s_0 + V - r) \\
> & c_h[mV + (q - m)(s_0 + V) - \frac{(q-m)(m+q+1)r}{2}] + c_h \frac{(q-m)(q+1-m)r}{2} - c_h(q - m)(s_0 + V - r) \\
= & c_h r(q + m^2 - m) \\
> & 0.
\end{aligned}$$



Therefore,  $\Delta_T(i_1, q+1) > \Delta_T(i_1, q)$ . By repeatedly using (62) and (63) as we keep adding  $w$  by 1 every time, we conclude that  $a_p^* = (i_1, u)$ .

**Case 3:  $s_0 \geq V$ .**

Suppose  $kV \leq mr \leq s_0 < (m+1)r \leq (k+1)V$  where  $k, m$  are positive integers. Since  $V = qr$ ,  $kq \leq m < kq+q$ . Let  $m = kq+k'$ . As in Case 1 and 2, (62) still follows when  $l_R > 0$ . Since the system has stockout starting from the  $(k+1)$ -th production cycle,  $l_D \geq k$ . When  $l_D > k$ , by the same reasoning given in case 1, (63) still follows. When  $l_D = k$ , since  $\Delta_T(i_1, (k+1)q) > 0$ , we have

$$\begin{aligned}
& \Delta_T(i_1, (k+1)q) \\
&= \Lambda_T(i_1, (k+1)q) + \Phi_T(i_1, (k+1)q) \\
&= c_s \sum_{\tau=kq+k'+1}^{kq+q} (\tau r - s_0) - c_h [\sum_{\tau=1}^k \tau q V + k'(k+1)V + \sum_{\tau=kq+k'+1}^{kq+q} (s_0 + V - \tau r)] \\
&= c_s \left[ \frac{(2kq+k'+q+1)(q-k')r}{2} - (q-k')s_0 \right] \\
&\quad - c_h \left[ \frac{k(k+1)}{2} qV + k'(k+1)V + (q-k')(s_0 + V) - \frac{(2kq+k'+q+1)(q-k')r}{2} \right] \\
&> 0.
\end{aligned}$$

Hence, by applying the above result in the first inequality below, we obtain

$$\begin{aligned}
& (q-k')[\Delta_T(i_1, kq+q+1) - \Delta_T(i_1, kq+q)] \\
&= c_s [(kq+q+1)(q-k')r - (q-k')s_0] - c_h (q-k')(s_0 + V - r) \\
&= c_s \left[ \frac{(2kq+k'+q+1)(q-k')r}{2} - (q-k')s_0 \right] + c_s \frac{(q-k')(q-k'+1)r}{2} - c_h (q-k')(s_0 + V - r) \\
&> c_h \left[ \frac{k(k+1)}{2} qV + k'(k+1)V + (q-k')(s_0 + V) - \frac{(2kq+k'+q+1)(q-k')r}{2} \right] \\
&\quad + c_h \frac{(q-k')(q-k'+1)r}{2} - c_h (q-k')(s_0 + V - r) \\
&= c_h r \left[ kq \left( \frac{1}{2}kq - \frac{1}{2}q + 2k' \right) + q + k'^2 - k' \right] \\
&> 0.
\end{aligned}$$

Therefore,  $\Delta_T(i_1, kq+q+1) > \Delta_T(i_1, kq+q)$ . By repeatedly using (62) and (63) as we keep adding  $w$  by 1 every time, we conclude that  $a_p^* = (i_1, u)$ .

Therefore, For any stationary production plan, the worst-case scenario  $a_p^* \in \{(s, 0), (s, u), s = i_1, i_2, \dots, i_m\}$ .  $\square$

### 4.3.3 Robust optimal solutions

In this section, we prove that there exists a stationary production plan in set  $\mathcal{X}$  that is optimal to the robust optimization problem.

**Proposition 5** *There exists a stationary production plan in set  $\mathcal{X}$  that is optimal to (RO).*

*Proof:* Let  $p = (i_1, i_2, \dots, i_m) \notin \mathcal{X}$ . Assume  $s_0 = \min_{\tau=1}^m I_{i_{\tau-1}}^+$  is the lowest inventory level prior to a replenishment in production plan  $p$ , and  $m_0 = \operatorname{argmin}_{\tau=1}^m I_{i_{\tau-1}}^+$ . Let  $p' = (i'_1, i'_2, \dots, i'_m)$  be a stationary production plan in set  $\mathcal{X}$ , in which a replenishment occurs once the inventory level is  $s_0$ . Suppose  $a_{p'}^*$  is the worst-case scenario for production plan  $p'$ . Given Proposition 4, there are two cases:

**Case 1:  $a_{p'}^*$  has no disruption.**

Since the planned inventory cost of  $p$  is higher than that of  $p'$ , and no stockout occurs, i.e.,  $\Delta_T(p, a_{p'}^*) = \Delta_T(p', a_{p'}^*) = 0$ , we have  $\Pi(p', a_{p'}^*) < \Pi(p, a_{p'}^*)$ . Combining the fact that  $\Pi^*(p') = \Pi(p', a_{p'}^*)$  and  $\Pi(p, a_{p'}^*) \leq \Pi^*(p)$ , we conclude that  $\Pi^*(p') < \Pi^*(p)$ . Therefore,  $p$  is not optimal.

**Case 2:  $a_{p'}^* = (i'_1, u)$ .**

Let  $a = (i_{m_0}, u)$  be a disruption for plan  $p$ . Since the planned inventory level under plan  $p'$  at period  $i'_1$  equals the planned inventory level under plan  $p$  at period  $m_0$ , the stockout cost of plan  $p$  under scenario  $a$  equals the stockout cost of plan  $p'$  under scenario  $a_{p'}^*$ , i.e.  $\Lambda_T(p, a) = \Lambda_T(p', a_{p'}^*)$ . Since the inventory is always back to the scheduled level after a disruption, the inventory levels of plan  $p$  under scenario  $a$  are at least as high as those of plan  $p'$  under scenario  $a_{p'}^*$  for all periods. Therefore  $\Pi(p', a_{p'}^*) \leq \Pi(p, a_{p'}^*)$ . Combining the fact that  $\Pi^*(p') = \Pi(p', a_{p'}^*)$  and  $\Pi(p, a_{p'}^*) \leq \Pi^*(p)$ , we conclude that  $\Pi^*(p') \leq \Pi^*(p)$ , i.e.,  $p'$  is a solution at least as robust as  $p$ .

Therefore, there exists a stationary production plan in set  $\mathcal{X}$  that is optimal to (RO).  $\square$

In this section, we proposed an approach to find optimal solutions of the robust optimization problem. Our main result (Proposition 5) is achieved with two steps. First, we show that the worst-case scenario for a stationary production plan has either no disruptions or a single  $u$ -period consecutive disruption (Proposition 4). This conclusion is obtained by classifying the base stock levels of the stationary production plans. Secondly, for any non-stationary production plan, we construct a stationary production plan, and show that the worst-case scenario associated with the stationary production plan has a total cost not greater than the cost of the non-stationary production plan under the scenario. Therefore, we conclude that we can find an optimal solution in the set of stationary production plans.

When the uncertainty budget is  $u$ , there are  $u + 1$  stationary production plans to search through. For each stationary production plan, we only need to examine two cases to obtain its worst-case scenario. Therefore, we provide an efficient algorithm that runs in at most pseudo-polynomial time to solve the robust optimization problem.

#### ***4.4 Numerical experiments***

In this section, we provide numerical results for the robust optimization problem. Figures 14-16 show the total cost increase due to scenarios with a single consecutive machine disruption for stationary production plans. The horizontal axis represents the length of disruptions, and the vertical axis represents the cost increase, where negative values indicate that the disruptions actually lower the total costs. In Figure 14, we present the cost changes under various base-stock levels  $s_0$  with batch size  $V = 100$ , demand rate  $r = 10$ , unit stockout cost  $c_s = 3$  and unit holding cost  $c_h = 1$ . For each given  $s_0$ , the cost changes move from negative to positive values as the

length of disruptions increases, and we can lower the cost increase due to machine disruptions by keeping a high base-stock level. For each base-stock level, the stockout occurs starting from the 1st, 3rd, 5th, 7th and 9th disruption day, respectively, and the cost changes increase monotonically when they become positive. When the base-stock level  $s_0 = 85$ , since a consecutive 20-period disruption cannot cause a cost increase, the worst-case scenario has no disruptions if the uncertainty budget  $u = 20$ . In Figure 15, we show the cost changes under different unit stockout costs  $c_s$  with batch size  $V = 100$ , demand rate  $r = 10$ , base-stock level  $s_0 = 60$  and unit holding cost  $c_h = 1$ . We observe that stockout occurs starting from the 7th disruption day for all the five curves; however, the cost changes increase by various rates as the length of disruptions increases. In Figure 16, we give the cost changes under different demand rates  $r$  with batch size  $V = 300$ , base-stock level  $s_0 = 150$ , unit stockout cost  $c_s = 1.5$  and unit holding cost  $c_h = 1$ . For the cases with demand rate  $r = 25$  and 30, a cost increase occurs when the length of the disruption increases to 15 and 10 periods, respectively. When demand rate  $r = 10, 15$  and 20, the worst-case scenario has no disruptions if the uncertainty budget  $u = 20$ . All of these results are aligned with the conclusion shown in Proposition 4.

As we observe in Figure 14, we can lower the cost increase due to machine disruptions by keeping a high base-stock level. However, by doing that we increase the holding cost in the absence of disruptions. Since optimal solutions to the robust model are stationary production plans, such holding cost increase stays the same for each production cycle. Robust solutions are usually attractive when the stockout cost are much higher than the holding cost. In Figure 17, we show an example with planning horizon  $|\mathcal{T}| = 100$ , unit stockout cost  $c_s = 30$ , unit holding cost  $c_h = 1$ , batch size  $V = 100$  and demand rate  $r = 10$ . We compare the three production plans with different base-stock levels under uncertainty budget  $u$  varying from 0 to 10. The vertical axis represents the total costs under the worst-case scenarios. We observe

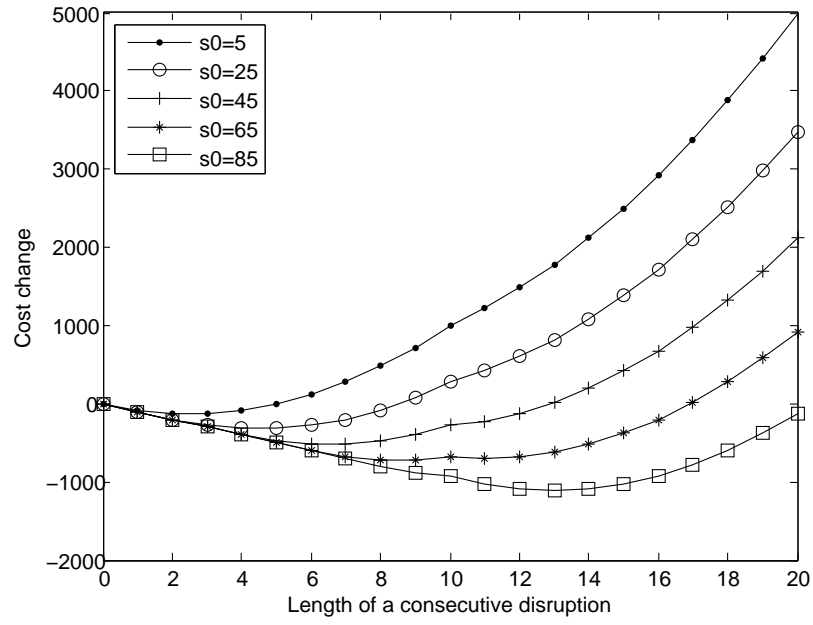


Figure 14: Cost changes due to machine breakdowns under different base stock levels

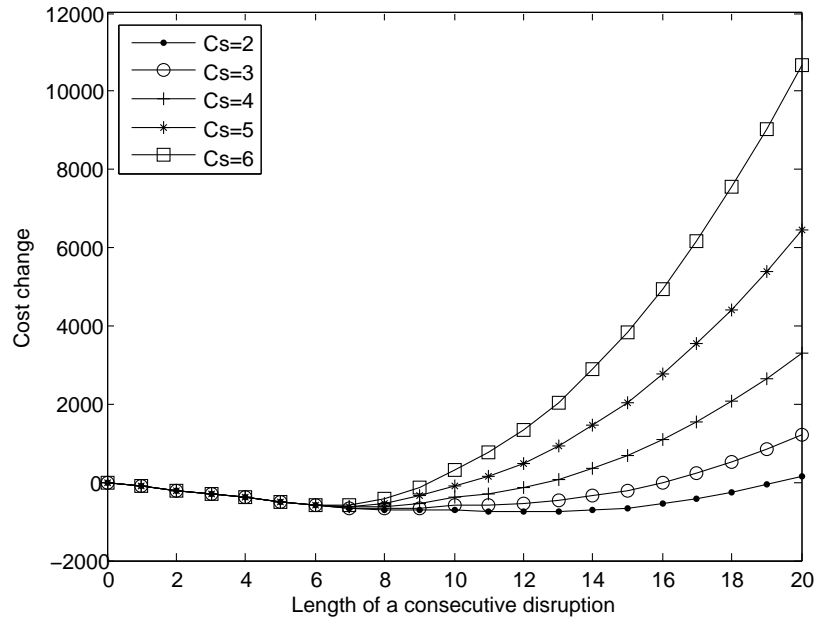


Figure 15: Cost changes due to machine breakdowns under different stockout rates

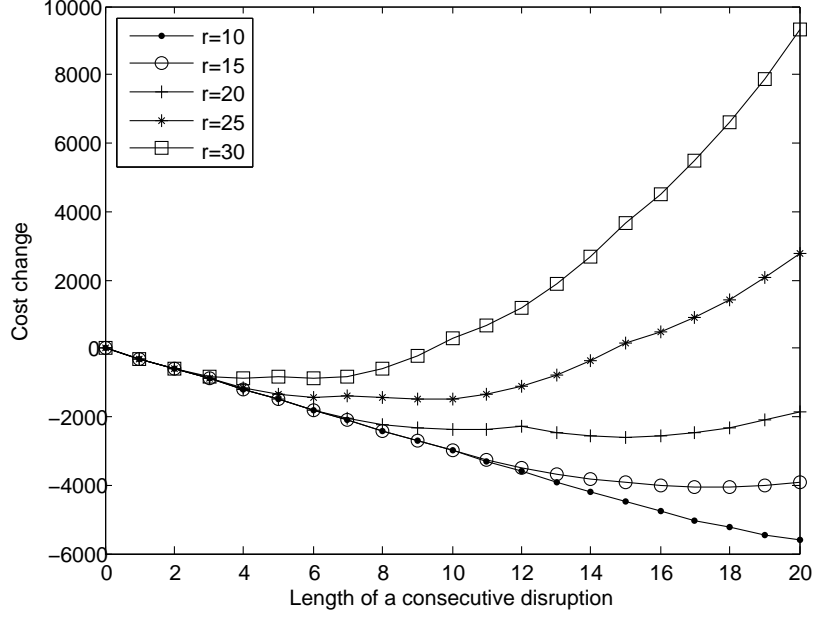


Figure 16: Cost changes due to machine breakdowns under different demand rates

that when the machine might fail to work in 4% periods of the planning horizon, the solution with base-stock level  $s_0 = 5$  is not optimal among the three cases. When the failure rate reaches 5%, the solution with base-stock level  $s_0 = 25$  performs the best under the worst-case scenario.

The problem considered in this chapter is related to maritime inventory management concerned with supply uncertainty. For a port with a stable consumption rate of a single product but subject to supply uncertainty, keeping a high base stock level can lower the cost increase in the presence of disruptions. The disruptions that we are concerned about can delay the arrivals of the product so as to lower the inventory level, and significantly increase the risk of stockout. We show that under the assumptions in this study, scheduling a delivery every time the inventory reaches a certain threshold level is an optimal solution of our robust model against the worst-case scenarios. Such strategy is especially attractive when the probability of disruptions is non-negligible and a high stockout cost is involved.

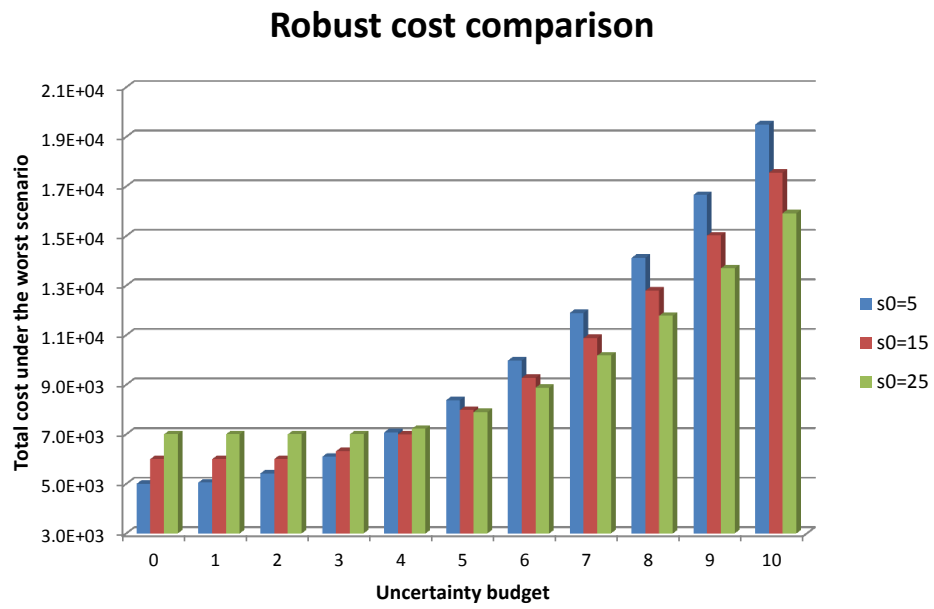


Figure 17: Robust cost comparisons for various base-stock levels and uncertainty budgets

## CHAPTER V

### CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Compared with other modes of transport, the potential of robust scheduling in the sector of maritime transportation has remained largely untapped. As a primary contribution of this thesis, we develop robustness planning strategies for an important class of optimization problems under uncertainty in maritime transportation. Such problems arise in the creation and negotiation of long-term delivery contracts with customers who require on-time deliveries of high-value goods throughout the year. We are concerned with dealing with some major types of disruptions that increase travel times between ports and ultimately affect one or more scheduled deliveries to the customers. In this section, we summarize the approaches and results presented in the thesis, and discuss some future research directions.

Since the final delivery contracts between a vendor and its customers are reached through many rounds of negotiations and discussions, at each iteration of this process and after the final agreements are determined, the vendor must generate routing solutions according to the tentative (or final) agreements to check their feasibility, and examine the operational costs and the robustness of the voyages. Motivated by this need, in Chapter II, we study a MIRP with given time windows for deliveries under random disruptions. The objective is to find robust solutions that can withstand unplanned disruptions. We propose a Lagrangian heuristic algorithm for obtaining robust solutions by introducing auxiliary soft constraints that are incorporated in the objective function with Lagrange multipliers. To evaluate the flexibility of solutions,



we build a simulator that generates disruptions and recovery solutions. Computational results show that by incurring a small increase in initial cost (sometimes zero), our robust planning strategies generate solutions that are often significantly less vulnerable to potential disruptions. We also consider the effect of lead time in being able to respond to the disruptions.

In Chapter III, we consider a more general MIRPTW where the length and placement of the time windows are also decision variables. We cast the problem as a two-stage stochastic MIP and propose a two-phase solution approach that considers a sample set of disruptions as well as their recovery solutions. In the first phase, we introduce two planning strategies to generate robust routes in which time buffers are spread among deliveries and consecutive deliveries at a port are separated by at least some minimum number of periods. In the second phase, we propose a multi-scenario construction heuristic to obtain good feasible solutions. Computational results reveal that our integrated solution procedure with judicious placement of time buffers and committed time windows leads to robust solutions that are less vulnerable to unplanned disruptions and have lower expected costs. We also investigate an iterative procedure between updating the routes and re-optimizing the time windows by coupling the Lagrangian heuristic approach proposed in Chapter II.

In Chapter IV, we study a robust single-item uncapacitated lot-sizing problem with backlogging and random machine breakdowns. By assuming an uncertainty budget of machine disruptions, we adapt the traditional notion of robust optimization to minimize total costs against the worst-case scenario. We reveal the solution structures under the worst-case scenarios, and show that the optimal solutions to the robust model can be characterized by a set of stationary production plans.

This research can be extended along several dimensions. One possibility is exploring other sources of uncertainty such as production and demand fluctuations, and price changes in spot markets. For instance, we can take advantage of a high price

spot market by sending a vessel there to sell products if our schedule has such flexibility. One way of achieving this is by scheduling long periods of idle time for one or multiple vessels in the original schedules. We may also extend the research to the cases where multiple disruptions are considered in a scenario. Although a multi-stage model significantly increases computational effort, it is a better approximation of the true problem. To overcome such computational issues, we can explore various methods of aggregating the scenarios of disruptions. Furthermore, future studies can also be devoted to enhancing the formulations in order to solve larger instances of the problems.

## REFERENCES

- [1] ABBOUD, N. E., “A simple approximation of the emq model with poisson machine failures,” *Production Planning & Control*, vol. 8, no. 4, pp. 385–397, 1997.
- [2] ABBOUD, N., “A discrete-time markov production-inventory model with machine breakdowns,” *Computers & Industrial Engineering*, vol. 39, no. 1, pp. 95–107, 2001.
- [3] ACHTERBERG, T., KOCH, T., and MARTIN, A., “Branching rules revisited,” *Operations Research Letters*, vol. 33, no. 1, pp. 42–54, 2005.
- [4] AGEEVA, Y., *Approaches to incorporating robustness into airline scheduling*. PhD thesis, Massachusetts Institute of Technology, 2000.
- [5] AGRA, A., CHRISTIANSEN, M., FIGUEIREDO, R., HVATTUM, L. M., POSS, M., and REQUEJO, C., “Layered formulation for the robust vehicle routing problem with time windows,” in *Combinatorial Optimization* (MAHJOUR, A. R., MARKAKIS, V., MILIS, L., and PASCHOS, V. T., eds.), pp. 249–260, Springer, 2012.
- [6] AGRA, A., CHRISTIANSEN, M., FIGUEIREDO, R., HVATTUM, L. M., POSS, M., and REQUEJO, C., “The robust vehicle routing problem with time windows,” *Computers & Operations Research*, vol. 40, no. 3, pp. 856–866, 2013.
- [7] ALVAREZ, J. F., TSILINGIRIS, P., ENGBRETHSEN, E. S., and KAKALIS, N. M., “Robust fleet sizing and deployment for industrial and independent bulk ocean shipping companies,” *INFOR: Information Systems and Operational Research*, vol. 49, no. 2, pp. 93–107, 2011.
- [8] ANDERSSON, H., CHRISTIANSEN, M., and FAGERHOLT, K., “Transportation planning and inventory management in the lng supply chain,” in *Energy, natural resources and environmental economics* (BJØRNDAL, E., BJØRNDAL, M., PARDALOS, P. M., and RÖNNQVIST, M., eds.), pp. 427–439, Springer, 2010.
- [9] ANDERSSON, H., HOFF, A., CHRISTIANSEN, M., HASLE, G., and LØKKETANGEN, A., “Industrial aspects and literature survey: Combined inventory management and routing,” *Computers & Operations Research*, vol. 37, no. 9, pp. 1515–1536, 2010.
- [10] BEN-TAL, A., EL GHAOU, L., and NEMIROVSKI, A., *Robust optimization*. Princeton University Press, 2009.

- [11] BEN-TAL, A., BERTSIMAS, D., and BROWN, D. B., “A soft robust model for optimization under ambiguity,” *Operations research*, vol. 58, no. 4-part-2, pp. 1220–1234, 2010.
- [12] BEN-TAL, A., GORYASHKO, A., GUSLITZER, E., and NEMIROVSKI, A., “Adjustable robust solutions of uncertain linear programs,” *Mathematical Programming*, vol. 99, no. 2, pp. 351–376, 2004.
- [13] BERTHOLD, T., “Primal heuristics for mixed integer programs,” 2006.
- [14] BERTSIMAS, D., BROWN, D. B., and CARAMANIS, C., “Theory and applications of robust optimization,” *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.
- [15] BERTSIMAS, D. and SIM, M., “The price of robustness,” *Operations research*, vol. 52, no. 1, pp. 35–53, 2004.
- [16] CACCHIANI, V., CAPRARA, A., and FISCHETTI, M., “A lagrangian heuristic for robustness, with an application to train timetabling,” *Transportation Science*, vol. 46, pp. 124–133, 2012.
- [17] CHAKRABORTY, T., GIRI, B. C., and CHAUDHURI, K., “Production lot sizing with process deterioration and machine breakdown,” *European Journal of Operational Research*, vol. 185, no. 2, pp. 606–618, 2008.
- [18] CHEUNG, K. L. and HAUSMAN, W. H., “Joint determination of preventive maintenance and safety stocks in an unreliable production environment,” *Naval Research Logistics (NRL)*, vol. 44, no. 3, pp. 257–272, 1997.
- [19] CHIRAPHADHANAKUL, V., *Routing and scheduling models for robust allocation of slack*. PhD thesis, Massachusetts Institute of Technology, 2010.
- [20] CHIU, S. W., “Robust planning in optimization for production system subject to random machine breakdown and failure in rework,” *Computers & Operations Research*, vol. 37, no. 5, pp. 899–908, 2010.
- [21] CHIU, S. W., WANG, S.-L., and CHIU, Y.-S. P., “Determining the optimal run time for epq model with scrap, rework, and stochastic breakdowns,” *European Journal of Operational Research*, vol. 180, no. 2, pp. 664–676, 2007.
- [22] CHIU, Y.-S. P., CHEN, K.-K., and TING, C.-K., “Replenishment run time problem with machine breakdown and failure in rework,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 1291–1297, 2012.
- [23] CHRISTIANSEN, M., FAGERHOLT, K., NYGREEN, B., and RONEN, D., “Maritime transportation,” in *Handbooks in Operations Research and Management Science*, pp. 189–284, Elsevier, 2007.
- [24] CHRISTIANSEN, M., FAGERHOLT, K., and RONEN, D., “Ship routing and scheduling: Status and perspectives,” *Transportation Science*, vol. 38, pp. 1–18, 2004.

- [25] CHRISTIANSEN, M., “Decomposition of a combined inventory and time constrained ship routing problem,” *Transportation Science*, vol. 33, no. 1, pp. 3–16, 1999.
- [26] CHRISTIANSEN, M. and FAGERHOLT, K., “Robust ship scheduling with multiple time windows,” *Naval Research Logistics (NRL)*, vol. 49, no. 6, pp. 611–625, 2002.
- [27] CHRISTIANSEN, M., FAGERHOLT, K., NYGREEN, B., and RONEN, D., “Ship routing and scheduling in the new millennium,” *European Journal of Operational Research*, vol. 228, no. 3, pp. 467–483, 2013.
- [28] CHRISTIANSEN, M. and NYGREEN, B., “A method for solving ship routing problems with inventory constraints,” *Annals of Operations Research*, vol. 81, pp. 357–378, 1998.
- [29] CHUNG, K.-J., “Bounds for production lot sizing with machine breakdowns,” *Computers & industrial engineering*, vol. 32, no. 1, pp. 139–144, 1997.
- [30] CHUNG, K.-J., “Approximations to production lot sizing with machine breakdowns,” *Computers & operations research*, vol. 30, no. 10, pp. 1499–1507, 2003.
- [31] COELHO, L. C., CORDEAU, J.-F., and LAPORTE, G., “Thirty years of inventory routing,” *Transportation Science*, vol. 48, no. 1, pp. 1–19, 2013.
- [32] FAGERHOLT, K., “Ship scheduling with soft time windows: An optimisation based approach,” *European Journal of Operational Research*, vol. 131, no. 3, pp. 559–571, 2001.
- [33] FISCHETTI, M. and MONACI, M., “Light robustness,” in *Robust and online large-scale optimization*, pp. 61–84, Springer, 2009.
- [34] FISHER, M. L., “The lagrangian relaxation method for solving integer programming problems,” *Management science*, vol. 50, no. 12-supplement, pp. 1861–1871, 2004.
- [35] GIRI, B. and DOHI, T., “Exact formulation of stochastic EMQ model for an unreliable production system,” *Journal of the Operational Research Society*, vol. 56, no. 5, pp. 563–575, 2005.
- [36] GOEL, V., FURMAN, K. C., SONG, J.-H., and EL-BAKRY, A. S., “Large neighborhood search for LNG inventory routing,” *Journal of Heuristics*, vol. 18, no. 6, pp. 821–848, 2012.
- [37] GROENEVELT, H., PINTELON, L., and SEIDMANN, A., “Production batching with machine breakdowns and safety stocks,” *Operations Research*, vol. 40, no. 5, pp. 959–971, 1992.

- [38] GROENEVELT, H., PINTELON, L., and SEIDMANN, A., “Production lot sizing with machine breakdowns,” *Management Science*, vol. 38, no. 1, pp. 104–123, 1992.
- [39] GRØNHAUG, R. and CHRISTIANSEN, M., “Supply chain optimization for the liquefied natural gas business,” in *Innovations in distribution logistics* (NUNEN, J. A., SPERANZA, M. G., and BERTAZZI, L., eds.), pp. 195–218, Springer, 2009.
- [40] GRØNHAUG, R., CHRISTIANSEN, M., DESAULNIERS, G., and DESROSIERS, J., “A branch-and-price method for a liquefied natural gas inventory routing problem,” *Transportation Science*, vol. 44, no. 3, pp. 400–415, 2010.
- [41] HALVORSEN-WEARE, E. E., FAGERHOLT, K., and RÖNNQVIST, M., “Vessel routing and scheduling under uncertainty in the liquefied natural gas business,” *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 290–301, 2013.
- [42] HAN, J., LEE, C., and PARK, S., “A robust scenario approach for the vehicle routing problem with uncertain travel times,” *Transportation Science*, 2013.
- [43] HARRIS, F. W., “How many parts to make at once,” *Operations Research*, vol. 38, no. 6, pp. 947–950, 1990.
- [44] HELD, M., WOLFE, P., and CROWDER, H. P., “Validation of subgradient optimization,” *Mathematical programming*, vol. 6, pp. 62–88, 1974.
- [45] JABALI, O., LEUS, R., VAN WOENSEL, T., and DE KOK, T., “Self-imposed time windows in vehicle routing problems,” *OR Spectrum*, pp. 1–22, 2013.
- [46] KENYON, A. S. and MORTON, D. P., “Stochastic vehicle routing with random travel times,” *Transportation Science*, vol. 37, no. 1, pp. 69–82, 2003.
- [47] KIM, C. H., HONG, Y., and KIM, S.-Y., “An extended optimal lot sizing model with an unreliable machine,” *Production Planning & Control*, vol. 8, no. 6, pp. 577–585, 1997.
- [48] KUHN, H., “A dynamic lot sizing model with exponential machine breakdowns,” *European Journal of Operational Research*, vol. 100, no. 3, pp. 514–536, 1997.
- [49] LAMBERT, V., LAPORTE, G., and LOUVEAUX, F., “Designing collection routes through bank branches,” *Computers & Operations Research*, vol. 20, no. 7, pp. 783–791, 1993.
- [50] LAN, S., CLARKE, J.-P., and BARNHART, C., “Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions,” *Transportation Science*, vol. 40, pp. 15–28, 2006.
- [51] LAPORTE, G., LOUVEAUX, F., and MERCURE, H., “The vehicle routing problem with stochastic travel times,” *Transportation science*, vol. 26, no. 3, pp. 161–170, 1992.

- [52] LIEBCHEN, C., LÜBBECKE, M., MÖHRING, R., and STILLER, S., “The concept of recoverable robustness, linear programming recovery, and railway applications,” in *Robust and online large-scale optimization*, pp. 1–27, Springer, 2009.
- [53] LIN, G. C. and GONG, D.-C., “On a production-inventory system of deteriorating items subject to random machine breakdowns with a fixed repair time,” *Mathematical and Computer Modelling*, vol. 43, no. 7, pp. 920–932, 2006.
- [54] LIU, B. and CAO, J., “Analysis of a production–inventory system with machine breakdowns and shutdowns,” *Computers and Operations Research*, vol. 26, no. 1, pp. 73–91, 1999.
- [55] MAKIS, V. and FUNG, J., “An emq model with inspections and random machine failures,” *Journal of the Operational Research Society*, vol. 49, no. 1, pp. 66–76, 1998.
- [56] MOINZADEH, K. and AGGARWAL, P., “Analysis of a production/inventory system subject to random disruptions,” *Management Science*, vol. 43, no. 11, pp. 1577–1588, 1997.
- [57] NEMHAUSER, G. L. and WOLSEY, L. A., *Integer and combinatorial optimization*, vol. 18. Wiley New York, 1988.
- [58] PAPAGEORGIOU, D. J., CHEON, M.-S., KEHA, A. B., NEMHAUSER, G. L., and SOKOL, J., “Mirplib: A maritime inventory routing survey and instance library.” Georgia Institute of Technology, 2012.
- [59] PAPAGEORGIOU, D. J., KEHA, A. B., NEMHAUSER, G. L., and SOKOL, J., “Two-stage decomposition algorithms for single product maritime inventory routing,” *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 825–847, 2014.
- [60] PAPAGEORGIOU, D. J., NEMHAUSER, G. L., SOKOL, J., CHEON, M.-S., and KEHA, A. B., “Mirplib—a library of maritime inventory routing problem instances: Survey, core model, and benchmark results,” *European Journal of Operational Research*, vol. 235, no. 2, pp. 350–366, 2014.
- [61] PETER CHIU, Y.-S., CHEN, K.-K., CHENG, F.-T., and WU, M.-F., “Optimization of the finite production rate model with scrap, rework and stochastic machine breakdown,” *Computers & mathematics with applications*, vol. 59, no. 2, pp. 919–932, 2010.
- [62] POCHET, Y. and WOLSEY, L. A., *Production planning by mixed integer programming*. Springer, 2006.
- [63] POSNER, M. and BERG, M., “Analysis of a production-inventory system with unreliable production facility,” *Operations Research Letters*, vol. 8, no. 6, pp. 339–345, 1989.
- [64] PRÉKOPA, A., *Stochastic programming*. Springer, 1995.

- [65] RAKKE, J. G., ANDERSSON, H., CHRISTIANSEN, M., and DESAULNIERS, G., “A new formulation based on customer delivery patterns for a maritime inventory routing problem,” *Transportation Science*, 2014.
- [66] RAKKE, J. G., STÅLHANE, M., MOE, C. R., CHRISTIANSEN, M., ANDERSSON, H., FAGERHOLT, K., and NORSTAD, I., “A rolling horizon heuristic for creating a liquefied natural gas annual delivery program,” *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 5, pp. 896–911, 2011.
- [67] ROSENBERGER, J. M., JOHNSON, E. L., and NEMHAUSER, G. L., “Rerouting aircraft for airline recovery,” *Transportation Science*, vol. 37, pp. 408–421, 2003.
- [68] ROSENBERGER, J. M., JOHNSON, E. L., and NEMHAUSER, G. L., “A robust fleet-assignment model with hub isolation and short cycles,” *Transportation Science*, vol. 38, pp. 357–368, 2004.
- [69] SCHAEFER, A. J., JOHNSON, E. L., KLEYWEGT, A. J., and NEMHAUSER, G. L., “Airline crew scheduling under uncertainty,” *Transportation Science*, vol. 39, pp. 340–348, 2005.
- [70] SHAPIRO, A., DENTCHEVA, D., and RUSZCZYŃSKI, A., *Lectures on stochastic programming: modeling and theory*, vol. 9. Society for Industrial and Applied Mathematics, 2009.
- [71] SHEBALOV, S. and KLABJAN, D., “Robust airline crew pairing: Move-up crews,” *Transportation Science*, vol. 40, pp. 300–312, 2006.
- [72] SHEN, Z., ORDÓÑEZ, F., and DESSOUKY, M. M., “The stochastic vehicle routing problem for minimum unmet demand,” in *Optimization and logistics challenges in the enterprise*, pp. 349–371, Springer, 2009.
- [73] SMITH, B. C. and JOHNSON, E. L., “Robust airline fleet assignment: Imposing station purity using station decomposition,” *Transportation Science*, vol. 40, pp. 497–516, 2006.
- [74] SONG, J.-H. and FURMAN, K. C., “A maritime inventory routing problem: Practical approach,” *Computers & Operations Research*, vol. 40, no. 3, pp. 657–665, 2013.
- [75] SPLIET, R. and GABOR, A. F., “The time window assignment vehicle routing problem,” *Transportation Science*, 2014.
- [76] STÅLHANE, M., RAKKE, J. G., MOE, C. R., ANDERSSON, H., CHRISTIANSEN, M., and FAGERHOLT, K., “A construction and improvement heuristic for a liquefied natural gas inventory routing problem,” *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 245–255, 2012.



- [77] TIRADO, G., HVATTUM, L. M., FAGERHOLT, K., and CORDEAU, J.-F., “Heuristics for dynamic and stochastic routing in industrial shipping,” *Computers & Operations Research*, vol. 40, no. 1, pp. 253–263, 2013.
- [78] WAGNER, H. M. and WHITIN, T. M., “Dynamic version of the economic lot size model,” *Management science*, vol. 5, no. 1, pp. 89–96, 1958.
- [79] YEN, J. W. and BIRGE, J. R., “A stochastic programming approach to the airline crew scheduling problem,” *Transportation Science*, vol. 40, pp. 3–14, 2006.
- [80] ZHANG, C., NEMHAUSER, G., SOKOL, J., CHEON, M.-S., and KEHA, A., “Flexible solutions to maritime inventory routing problems with delivery time windows,” 2013. Submitted for publication.